

## Projecte/Treball fi de carrera

**Estudi:** ETIG. Pla 2001

**Títol:** AutoGest, aplicació de gestió per a empresaris autònoms

**Alumne/a:** Sandra Viñas Comamala

**Director/Tutor:** Francesc Castro Villegas

**Departament:** IMA

**Àrea:** Llenguatge i sistemes informàtics

**Convocatòria** (mes/any): 09/2008

**Index.**

<b>1. Introducció .....</b>	<b>1</b>
1.2 Introducció general .....	1
1.3 Objectius .....	2
<b>2. Estudi de la viabilitat del sistema.....</b>	<b>4</b>
2.1 Establir l'abast del sistema .....	4
2.2 Estudi de la situació actual .....	5
2.3 Definició dels requisits del sistema.....	6
2.4 Estudi d'alternatives per la solució .....	11
<b>3. Anàlisi del sistema d'informació .....</b>	<b>13</b>
3.1 Definició del sistema.....	13
3.1.1 Determinar l'abast del sistema.....	13
3.1.2 Identificació de l'entorn tecnològic .....	17
3.1.3 Especificació d'estàndards i normes.....	17
3.2 Establiment de requisits .....	18
3.2.1 Obtenció dels requisits del Sistema.....	18
3.2.2 Especificació de casos d'ús .....	24
3.3 Anàlisi dels casos d'ús.....	33
3.4 Anàlisi de classes.....	39
3.5 Definició d'interfície d'usuari .....	62
<b>4. Disseny del sistema d'informació.....</b>	<b>82</b>
4.1 Definició de l'arquitectura del sistema .....	82
4.1.1 Definició de nivells d'arquitectura.....	82
4.1.2 Identificació de requisits de disseny i construcció.	83
4.1.3 Especificació normes de disseny i contrucció.....	83

4.1.4 Identificació de subsistemes de disseny .....	84
4.1.5 Especificació de l'entorn tecnològic.....	85
4.1.6 Especificació de requisits d'operació i seguretat....	85
4.2 Disseny de l'arquitectura de suport .....	85
4.3 Disseny de casos d'ús reals .....	86
4.4 Disseny de classes .....	110
4.5 Disseny físic de dades .....	133
4.5.1 Transformació del model de classes al model físic	133
4.6 Especificacions de construcció .....	143
<b>5 Construcció del sistema d'informació .....</b>	<b>145</b>
5.1 Preparació de l'entorn de generació de contrucció .....	145
5.2 Generació de codi de components .....	145
5.3 Elaboració de proves .....	146
5.4 Elaboració dels manuals d'usuari.....	146
5.5 Definició de la formació de l'usuari final .....	146
<b>6. Incorporació del sistema a l'entorn final .....</b>	<b>147</b>
6.1 Incorporació del sistema a l'entorn fila .....	147
6.2 Realització de proves d'implantació .....	147
6.3 Realització de proves d'acceptació del sistema .....	147
<b>Conclusions .....</b>	<b>148</b>
<b>Biografia .....</b>	<b>150</b>

# I Introducció.

## I.2 Introducció general.

El projecte de final de carrera és la última fase de la carrera universitària on s'ha de demostrar tots els coneixements que un ha adquirit al llarg de la formació acadèmica universitària i posar-los en pràctica. Aquest treball requereix molta dedicació i esforç, però que al final queda compensat amb il·lusió i satisfacció personal i professional.

L'origen del projecte va sorgir a mitjans de la meua vida universitària quan el meu pare, empresari autònom, va decidir que havia de renovar-se i començar a fer la facturació de manera informatitzada, de manera que deixava de banda el bolígraf i els blocs de factures i passava al teclat. Aquesta decisió va comportar que ell, per primera vegada, s'havia de posar davant d'un ordinador per poder fer la feina administrativa. Assessorat per altres empesaris que ja utilitzaven la informàtica per la gestió de la facturació ell va escollir un programa bastant fàcil d'utilitzar per un iniciat amb els ordinador, però degut a la seva inexperiència amb el camp informàtic el programa resultava poc intuïtiu d'utilitzar. Va ser en aquest moment que el meu pare va reclamar la meua ajuda per tal d'ensenyar-li com funcionava el programa, tot i saben la meua inexperiència administrativa. Després de sentir-lo sempre a renegar del programa li vaig prendre la decisió que el meu projecte de final de curs podria ser realitzar un programa de facturació totalment flexible per qualsevol tipus d'empresa petita (per autònoms) que el principal objectiu fos la facilitat d'utilització i la intuïció per poder realitzar les diferents tasques generals administratives.

Un cop acceptada la proposta d'aquest projecte la meua principal feina inicial va ser aprendre com funcionava la part administrativa d'una empresa. El principal aprenentatge va ser durant l'estiu passat quan vaig entrar d'administrativa a una empresa, que es dedica a la distribució de productes congelats. Aquí vaig aprendre la majoria de coses que havia de tenir en compte alhora de realitzar el projecte.

Aquest projecte ha intentat respondre sobretot a la necessitat de tenir recollides totes les dades necessàries d'una empresa petita, per la generació de factures i el control d'estoc que té, de manera que en un futur es puguin ampliar les funcionalitats per especialitzar el programa a una empresa de qualsevol activitat.

Per acabar, cal destacar que en el programa s'ha tingut especial atenció en el disseny de la interfície gràfica per tal que l'usuari intuïtivament sàpiga on ha d'accedir per realitzar les diferents tasques.

### **1.3 Objectius.**

L'objectiu principal que es pretén arribar amb aquest projecte és: el desenvolupament d'una aplicació que permeti realitzar la gestió mínima tasques administratives necessàries per una empresa petita.

Les principals característiques del projecte a desenvolupar són les següents.

- Obtenir una eina que permeti la gestió dels diferents objectes que interaccionen amb l'empresa, és a dir, la gestió dels clients, proveïdors, treballadors.
- La gestió dels articles que ofereix l'empresa en la venda. Des de guardar la informació especificat de l'article fins al control d'estoc del mateix.
- Gestió de la facturació a clients, des de poder realitzar un pressupost, generar una comanda, realitzar un albarà de lliurament de productes o serveis fets, fins a la realització de la facturació per els productes servits o serveis prestats.
- Gestió de les compres i comandes que es fan als proveïdors per tal de tenir les despeses controlades i el control de comandes que se li demanen.
- Gestió de l'estoc dels articles que té l'empresa de manera que es puguin realitzar canvis en el nivell d'estoc dels articles, en cas d'entrades o sortides que no siguin degudes a entregues. Poder consultar en tot moment quins són els articles amb estoc baix.

- Poder treure llistats de totes les dades entrades, fer consultes sobre les activitats dels clients i proveïdors, la quantitat d'articles comprats i venuts i per acabar el total d'ingressos i despeses de l'empresa.

## 2. Estudi de la viabilitat del sistema

En aquest capítol s'exposa l'estudi de viabilitat del sistema, on s'estudien les diferents activitats:

- Establir l'abast del sistema.
- Estudi de la situació actual.
- Definició de requisits del sistema.
- Estudi d'alternatives de solució
- Valoració de les alternatives.
- Selecció de la solució.

### 2.1 Establir l'abast del sistema.

Tota empresa petita en la seva activitat econòmica té la necessitat tenir un control de les despeses i ingressos que genera, és així que haurà de poder gestionar la facturació que genera, les despeses que realitza per poder portar a terme la seva activitat, la gestió del control d'estoc dels productes i la gestió dels ítems involucrats en cadascuna de les tasques esmentades anteriorment.

Analitzant la descripció del nou sistema proposat es pot observa que, com que ha de ser un sistema flexible per a qualsevol empresa, les restriccions en el moment de entrar dades i realització de les diferents activitats no poden ser molt restrictives, una de les coses que s'ha de tenir en compte es que el nivell de dades a gestionar no serà molt gran, ja que es tracta d'empreses petites.

Per tal de poder fer les diferents tasques administratives l'usuari haurà de ser l'encarregat d'entrar prèviament o durant totes les dades per poder-les realitzar. D'aquesta manera podrà tenir un control real del funcionament de l'empresa, tant econòmicament com en volum de treball i creixement empresarial.

L'estudi de la viabilitat del sistema comprendrà un estudi de quines són les necessitats de qualsevol possible usuari de l'aplicació de manera que no generi cap dificultat en la realització de les tasques normals que ha de fer.

## 2.2 Estudi de la situació actual.

Com que es tracta d'una aplicació que qualsevol usuari pugui utilitzar, no es pot saber quina és la situació actual en que es troba. Però partint que és un programa dirigit a empesaris amb empreses petites, podem suposar que en l'actualitat no tenen cap sistema informàtica més que una programa per els càlculs (fulla de càlcul) i una agenda.

Per tal de poder fer un estudi de la situació actual de com treballen els diferents possibles usuaris, me fixat amb la comunitat d'empesaris autònoms, ja que, cada vegada es més gran en el nostre país i són els que representen en majoria les empreses petites.

Com menciona ATA (Federació Nacional de Treballadors Autònoms) Espanya és un dels països amb menor inversió en el foment i l'ús de noves tecnologies, i es per això, que l'economia espanyola està encara en un procés de modernització, és a dir, una societat econòmica encara en desenvolupament de noves tecnologies d'informació i comunicació.

S'ha d'observar que la comunitat d'autònoms cada any augmenta, és així, que en el 2001 el nombre de treballadors autònoms eren als voltants dels 2.600.000 i el 2005 estava fregant els 3 milions de persones treballant com a tal, els qual representen un 16% de la comunitat treballadora segons indica el règim d'afiliació de la seguretat social.

Cal pensar que aquest col·lectiu forma empreses la majoria d'elles amb un sol treballador o pocs i la majoria creuen que les noves tecnologies en comptes de ajudar-los a facilitar la realització de les seves tasques els hi complicarà, ja que inicialment representa un aprenentatge i una posta a punt per poder treballa. Tot i això, cada vegada més molts empesaris pensen en informatitzar la seva empresa però a molts d'ells els ho impedeix els coneixements que tenen de la informàtica, els quals poden no haver tingut un primer contacte amb un ordinador.



Dels empesaris que realitzen la seva tasca administrativa de manera informàtica molts o fan amb una eina d'ofimàtica bàsica que ja porti incorporat l'ordinador.

## 2.3 Definició dels requisits del sistema.

Tot i que la societat actual està bastant avançada en el món de la informàtica, encara hi ha molta gent que té por a les noves tecnologies, i es per això, que tenint en compte aquesta peculiaritat, s'ha volgut realitzar una aplicació per tant fàcil d'utilitzar i amigable.

Aquest nou programari ha de ser un sistema de gestió de facturació per tal de agilitzar les tasques de gestió empresarial.

Abans de començar a parlar de les especificacions i requisits que ha de tenir l'aplicació s'explicaran un seguit de conceptes que s'han de tenir clars per tal d'entendre el que ha de fer l'aplicació i que és el que representa.

- Dades que ha de contenir el programa:
  - Client → Persona que usa habitualment els serveis d'un professional.
  - Proveïdor → Persona que té el càrrec de proveir d'una mercaderia o d'un producte una persona, una empresa.
    - Treballador → Persona que treballa en qualitat d'assalariat.
    - Article → Cada mena de coses comercials. En aquest cas article fa referència a productes i a serveis.
    - Família → Conjunt de persones o coses que es consideren del mateix origen o de les mateixes característiques. En aquest cas seran els articles que es classificaran en famílies.
    - Zones → Extensió de terreny limitada per raons administratives, comercials, polítiques, etc. En aquest cas l'activitat comercial de l'empresa pot ser dividida amb zones.
    - Entitat bancària → Banc que constitueix l'anomenat sistema bancari.

- Pressupost → Còmput anticipat de les despeses i els ingressos d'una unitat econòmica o entitat per a un període de temps determinat.
  - Comanda → Encàrrec fet per un client a un proveïdor de servir-li una mercaderia, un article, un producte o un servei determinats, d'una qualitat i unes característiques prèviament establertes, en un termini fixat i a un preu i en unes condicions convinguts.
  - Albarà → Document que normalment acompanya la mercaderia en ésser lliurada, i que n'indica la natura i la quantitat, la data del lliurament i el nom del comprador i del venedor, i que, si és conformat, acredita la recepció de la mercaderia.
  - Factura → Document que acompanya el lliurament de mercaderies o la prestació de serveis i en el qual hom en descriu les característiques i en fa constar el preu.
  - Devolució → Acció de tornar una cosa a la persona que la posseïa primerament.
  - Estoc → Quantitat d'una mercaderia que hi ha en magatzem.
- Paraules de tipus econòmic que s'usaran en el transcurs del projecte.
- Tarifes → Preu estipulat, presentat generalment en forma de taula o catàleg, per determinats serveis, drets a pagar, impostos, etc.
  - Descompte → Quantitat deduïda de l'import d'un deute o obligació
    - Per pagament immediat → Si es paga només d'entregar la comanda.
    - Comercial → Descompte que s'aplica a cada factura generada al client.
  - IVA → Impost sobre el valor afegit. Impost indirecte, de tipus multifàsic sobre les vendes, amb caràcter no acumulatiu sobre el consum, i que grava el valor afegit en cadascuna de les fases del procés productiu. Es calcula aplicant el tipus impositiu corresponent sobre la diferència entre el preu de venda i el preu de compra (valor afegit en la fase productiva), de

manera que és un impost neutral que, en la darrera transacció, recau totalment sobre el consumidor final.

- RE → Regim especial d'impost sobre el valor afegit aplicable a comerciants minoristes.
- IRPF → Impost sobre la renda de les persones físiques. impost directe i personal que grava la renda de les persones físiques ateses les seves circumstàncies personals i familiars.
- Risc no pagament → quantitat que una empresa pot suportar que una altre empresa o persona per no pagar.
- Recàrrec financer → impost que s'aplica en cas de pagar una factura fora del plas establert.

Com que el programari pot ser utilitzat per qualsevol tipus d'empresa petita o mitjana s'ha de pensar que no pot ser molt restrictiu, és per aquest motiu que poden haver apartats que depenent de quina empresa es sigui necessaris.

Hi ha un seguit d'operacions que són comunes a qualsevol empresa com ara:

- Guardar les dades informatives de: proveïdors, clients i .
- Realitzar documents com ara: Factures, albarans, comandes i pressupostos de clients.
- Tenir informació dels productes i serveis que s'ofereixen als clients
- Realitzar consultes sobre les vendes fetes.

Depenent de l'empresa també tindrà la necessitat de fer altres operacions com ara:

- Classificar articles en famílies.
- Guardar les diferents entitats bancàries que es treballa.
- Classificar els clients en zones geogràfiques.
- Fer un control d'estoc de productes.
- Gestionar les compres que es fan a proveïdors.
- Consulta de guanys i pèrdues.

Dites les diferents accions que ha de realitzar el software es passarà a especificar quines són les operacions que s'hauran de poder permetre fer el programa.

- Emmagatzemar les dades dels clients, proveïdors, treballadors, articles i entitats bancàries, de manera que s'hauran de poder entrar de nous, modificar-los, eliminar-los i consultar.
  - Cada client haurà de poder tenir associada a una zona, en cas que l'empresa treballi amb separació de zones i també se li haurà de poder assignar un treballador de tipus comercial o responsable en cas necessari. Si es treballa amb diferents tipus de tarifa, s'haurà de poder assignar una tarifa al client, de les quatre que s'hauran de poder posar, en cas de no treballar en diferents tarifes per defecte el programa haurà d'assignar la primera tarifa.
  - Els proveïdors son empreses externes que ofereixen a l'empresa diferents serveis com: proveir-nos de matèries primeres, articles per a la venda, serveis, producte per el manteniment, ...
  - Les persones que treballen per l'empresa, els treballadors, s'hauran de poder assignar a una o varies zones, en el cas que hi haguin zones definides.

- Els articles s'hauran de poder classificar en famílies i subfamílies. De cada article s'haurà de poder assignar un proveïdor habitual de la mateixa manera que se li hauran de poder assignar tots els proveïdors que ofereixen l'article o el servei.
- En una empresa hi ha un seguit de documents de venda que s'han de generar com són les comandes, factures, albarans i pressupostos. La informació que s'ha de guardar en comú d'aquests documents és el client a qui va dirigit el document, les dades de l'empresa, els articles que s'han comprat o demanat.
- Com que aquets documents uns estan relacionats amb els altres doncs s'haurà de poder fer el següent:
  - Les comandes es podran fer a partir d'un o varis pressupostos, ja realitzats anteriorment.
  - Els albarans es podran fer a partir d'un o varis pressupostos o comandes, ja fets amb anterioritat.
  - Les factures es podran fer a partir d'un o varis pressupostos, comandes o albarans, fets amb anterioritat.
- En cas que hi hagi una devolució en l'article el document d'albarà i factura hauran de ser en negatiu, de manera que un albarà és podrà modificar la quantitat, però la factura, s'haurà de generar una de nova amb import negatiu, ja que una factura no pot ésser modificada en el cas de ser validada.
- Una empresa pot tenir la necessitat de generar comandes per que el proveïdor les serveixi, és per això, que necessitarà generar una comandes per a proveïdors. Cal que només es pugui fer una comanda amb només els articles que el proveïdor serveix a l'empresa.
- Com que a l'empresa també arribaran factures dels diferents proveïdors, s'hauran de poder entrar per tal de tenir guardades les despeses que es generen per l'activitat empresarial.

- Per poder estar informats com funciona l'empresa interessarà treure llistats de qualsevol de les dades emmagatzemades, ingressos i despeses i facturació que es fa o que es rep.
- Fer consultes varies sobre els clients, proveïdors i articles.

## 2.4 Estudi d'alternatives per la solució

Si ens posem a pensar si no desenvolupéssim aquest podríem trobar moltes altres alternatives molt iguals i també més elaborades que aquesta.

Per començar tenim els mètodes més antics que són els manuals, és a dir, l'agenda amb els blocs d'albarans i factures, i algun full segellats més, llavors ja mirant en termes de software informàtic tenim les eines d'ofimàtica bàsica, com pot ser el Word o Excel que serien les substitutives del paper i el bolígraf sense utilitzar un sistema de gestió.

Llavors ja em de passar a parlar de programari que presenten altres empreses que n'hi ha molts per triar en el marcat, sobretot la facilitat d'obtenir-los és de casa, que amb Internet i una targeta de crèdit, amb un moment es té tot instal·lat i de la nostra propietat. El problema d'aquest serveis es que si actualitzen el software o el manteniment d'errors en alguns períodes, per utilitzar-los suposen un cost addicional cada any. També es poden obtenir programes a qualsevol botiga informàtica.

Que podem trobar en software ja generat per altres empreses.

Aquestes opcions poden ser totes al mateix paquet, com soles o combinables unes amb les altres.

- Agenda
- Gestió de personal
- Gestió d'atricsles.
- Gestió de magatzem.
- Comptabilitat.

- Facturació.
- Planificació de pressupostos i projectes.

Perquè el nou software que volem realitzar i no un altre?

Sobretot per que molts d'aquests programes són complicats d'utilitzar o entendre com funcionen fins que no s'hi ha treballat durant una bona temporada.

### 3 Anàlisi del sistema d'informació

En aquest capítol s'exposa l'anàlisi del sistema d'informació, segons el que indica l'apartat de Mètrica v.3 que té el mateix nom. En aquest apartat, per un anàlisi orientat a objectes. La Mètrica v.3 proposa estudiar les següents activitats:

- Definició del sistema
- Establiment de requisits
- Identificació de subsistemes d'anàlisi
- Anàlisi dels casos d'ús
- Anàlisi de classes
- Definició d'interfícies d'usuari.

#### 3.1. Definició del sistema

##### 3.1.1 *Determinar l'abast del sistema*

De tal manera de poder entendre millor l'abast del sistema, passarem a establir els casos d'ús a alt nivell, identificar per cada un l'actor principal i la relació d'ús entre els casos.

Cal tenir en compte que aquest programa a l'estar dirigit a petita i mitjana empresa només hi haurà un possible usuari, com pot ser el mateix empresari o un administratiu. Per tant només hi actuarà un actor.

Segons el model de negoci es pot extreure que el programa ha de tenir les següents funcionalitats.

- Gestió de la informació de l'empresa.
- Gestió dels clients.
- Gestió de proveïdors.
- Gestió de treballadors
- Gestió d'articles o serveis.
- Gestió de les zones.
- Gestió d'entitats bancàries



- Generació de comandes de clients.
- Realització de pressupostos.
- Gestió de lliurament de producte o servei.
- Gestió de cobrament.
- Gestió de devolucions.
- Gestió d'entrada de material.
- Gestió de pagaments.
- Gestió de devolucions.
- Resums d'ingressos i despeses i Consultes.
- Realització de comandes per a proveïdors.
- Consultes d'estoc baix.
- Gestió d'estoc.

Per tal de organitzar bé el programa i la implementació del mateix, passem a classificar les diferents funcionalitats en casos d'ús. El resultat són els següents casos i subcasos d'ús

### **Casos d'ús del sistema**

Actor principal: *Usuari*

- Gestió de dades físiques.
  - Gestió de la informació de l'empresa.
  - Gestió dels clients.
  - Gestió de proveïdors.
  - Gestió de treballadors
  - Gestió d'articles o serveis.
  - Gestió de les zones.
  - Gestió d'entitats bancàries
- Gestió de vendes.
  - Generació de comandes de clients.
  - Realització de pressupostos.
  - Gestió de lliurament de producte o servei.

- Gestió de cobrament.
- Gestió de devolucions.
- Gestió de compres.
  - Gestió d'entrada de material.
  - Gestió de pagaments.
  - Gestió de devolucions.
- Resums d'ingressos i despeses i Consultes.
- Control d'estocs de magatzem.
  - Realització de comandes per a proveïdors.
  - Consultes d'estoc baix.
  - Gestió d'estocs.

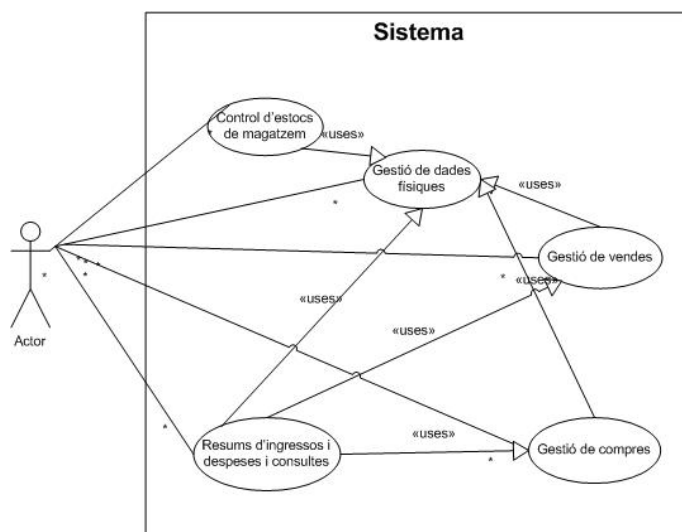
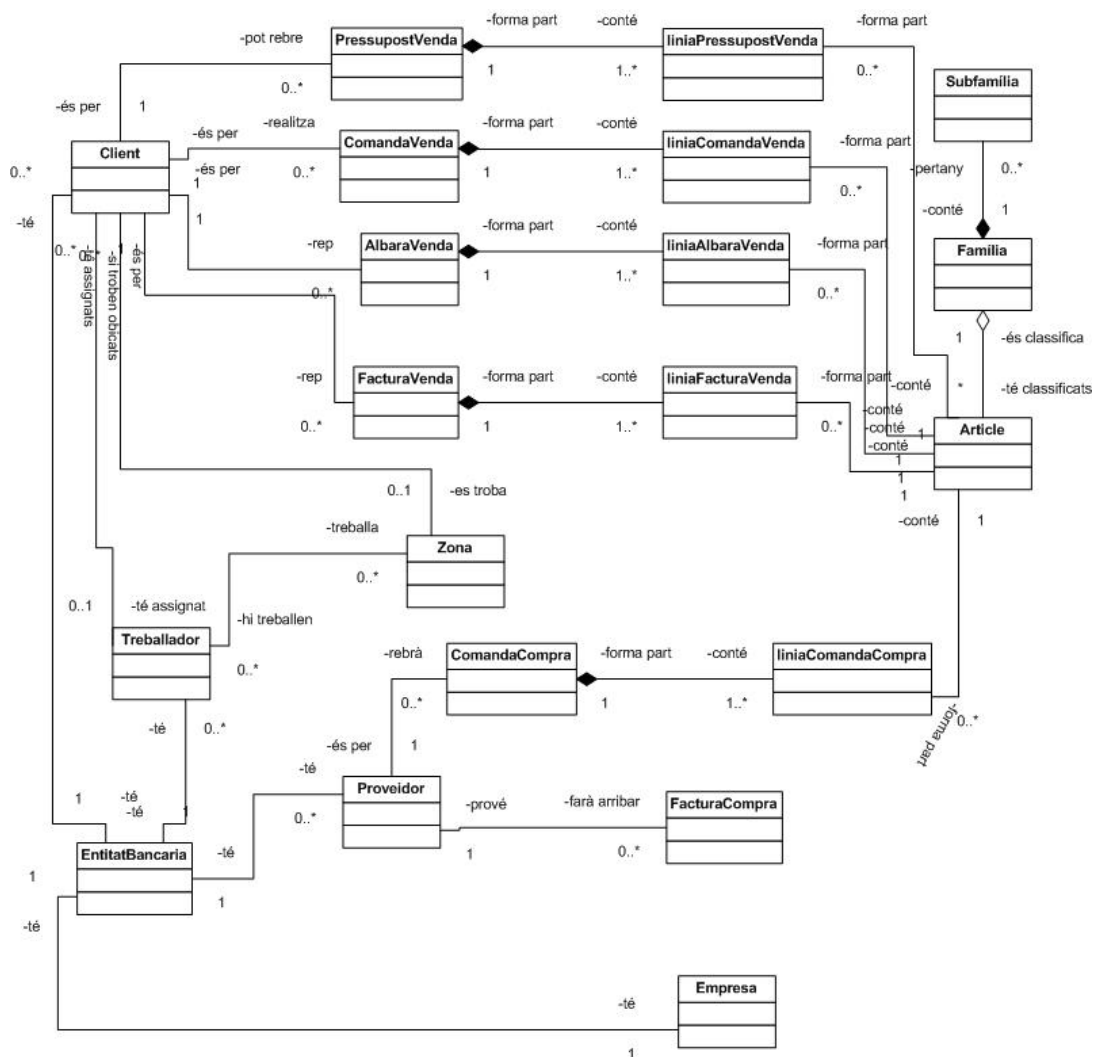


Diagrama de casos d'ús d'alt nivell

En l'estudi dels casos d'ús d'alt nivell s'identifiquen els objectes conceptuals que formen part del model de domini. Tot seguit es mostra el model.

Aquest model en fases d'anàlisi i de disseny s'anirà modificant.



### Model conceptual de domini

### **3.1.2 Identificació de l'Entorn tecnològic**

Per tal d'identificar l'entorn tecnològic i donar resposta al sistema, cal estudiar el model de negoci i de domini, a més de realitzar sessions de treball per tal de definir les restriccions tecnològiques que puguin aparèixer.

Cal tenir presents els requeriments tecnològics descrits en la definició de requeriments de l'apartat del Model de negoci, on posa que una de les condicions ha de ser que el programari addicional necessari haurà de ser, sense despesa econòmica i que qualsevol ordinador el pugui suportar. Es per això que el sistema de base de dades a utilitzar ha de ser el més estàndard possible, sempre i quan pugui complir amb els requeriments del programari. Es per això que una base de dades com MySQL la quan és robusta i lliure d'impostos, serà suficient per cobrir les necessitats.

L'altre requisit és que el sistema a construir pugui ser utilitzat sota un entorn Windows i amb el maquinari actual del que disposa qualsevol persona a casa, ja que és l'entorn normal de treball de qualsevol empresari.

Cal dir que el sistema ha de funcionar en un entorn d'un sol usuari i això comporta que no cal tenir en compte requeriments tecnològics de comunicacions. Tot el sistema, interfície i base de dades s'executarà, doncs des de la mateixa estació de treball.

### **3.1.3 Especificació d'estàndards i normes**

Respecte al compliment d'estàndards o normes, les fases d'anàlisi i disseny tenen en compte la nomenclatura de UML en la utilització de diagrames.

## **3.2 Establiment de requisits**

### **3.2.1 Obtenció dels requisits del Sistema**

#### ***Gestió de dades físiques***

##### ***Manteniment de l'empresa.***

Ha de permetre guardar la informació referent a l'empresa a la qual pertany el software, ja que aquesta és la informació que haurà de sortir en els documents com factures, albarans, pressupostos.

##### ***Gestió de client.***

Gestionar tota la informació referent al client, és a dir les dades, per la qual cosa s'haurà de poder entrar clients nous, poder modificar les dades en cas necessari, poder consultar la informació i poder donar-lo de baixa si s'escau.

##### ***Gestió de proveïdor.***

Com en el cas del client s'haurà de poder fer una gestió de la informació necessària de l'empresa proveïdora en qüestió de tal manera que es pugin entrar noves empreses, consultar-les, modificar-les si es necessari i poder-les donar de baixa.

##### ***Gestió de treballadors.***

En cas que l'empresa tingui treballadors d'aquets també se'n hauran de guardar les dades com el cas dels clients i proveïdors i fer les mateixes operacions de entrar-ne de nou, poder-los modificar, consultar-los i eliminar-los en cas que ja no treballin per l'empresa.

**Gestió d'articles.**

Tota empresa ofereix serveis o productes als seus clients, de tal manera que haurà tenir la informació de cadascun dels articles a vendre als clients i també si creu necessari els que compra als proveïdors. Per tant s'hauran de poder entrar, modificar, consultar i eliminar. Com que els articles poden estar classificats i subclassificats es farà amb famílies i subfamílies, per tant caldrà fer una gestió d'entrada d'aquestes ordenacions, per tant passarem a tenir un nou apartat per poder entrar, modificar, consultar i eliminar famílies i subfamílies.

**Gestió de zones.**

Com que els diferents clients i treballadors els podem tenir assignats a diferents zones també caldrà fer una gestió d'aquestes zones, així que s'haurà de poder entrar noves zones, modificar-les si cal, poder-les consultar i eliminar-les en cas de no necessitat.

**Gestió d'entitats bancàries .**

Com en el cas de proveïdors, clients i treballadors, una entitat bancària és una empresa o entitat física, que caldrà fer la mateixa gestió amb les dades necessàries.

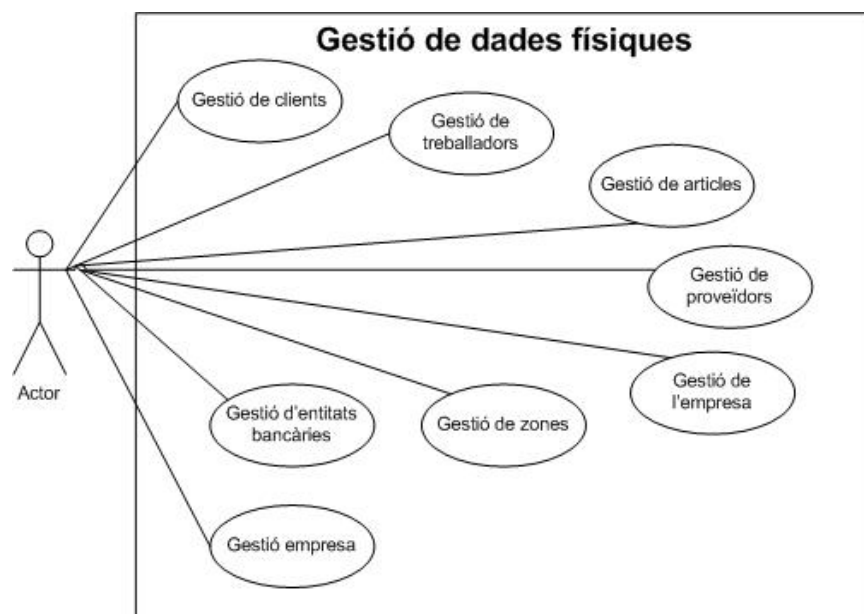


Diagrama de casos d'ús de Gestió de dades físiques

## ***Gestió de vendes***

### ***Introduir comandes dels clients***

Una comanda de venda serà per un client i constarà de diferents articles. La comanda la podrem entrar directe o fer-la a partir d'un pressupost del client que no s'hagi aprovat, de manera que quedarà aprovat un cop feta la comanda.

### ***Realitzar pressupostos***

Un pressupost de venda per un client i constarà de diferents articles. Aquest pressupost serà modificable sempre i quan no hagi estat aprovat.

### ***Gestionar lliurament de producte o servei***

La gestió de lliurament de producte es la creació d'albarans per els client. De manera que s'haurà de permetre fer albarans directament o a partir de pressupostos o comandes realitzades en anterioritat i que no hagin estat aprovades, és a dir, ja utilitzades per d'altres documents. Un cop aprovat un albarà voldrà dir que l'article ja ha estat entregat al client.

### ***Gestió de cobraments***

En aquest cas els cobraments es la realització de la factura del productes a entregar o entregats als clients.

Una factura es podrà realitzar a partir de comandes, pressupostos o albarans realitzats en anterioritat i que encara no hagin estat aprovats. També es podran entrar factures directament.

Un cop s'aprovi una factura voldrà dir que el client la pagada.

### ***Gestió de devolucions***

En aquest cas és la generació d'una factura negativa, la qual s'haurà d'entrar les dades amb negatiu. Un cop s'aprovi la factura voldrà dir que s'ha pagat al client.

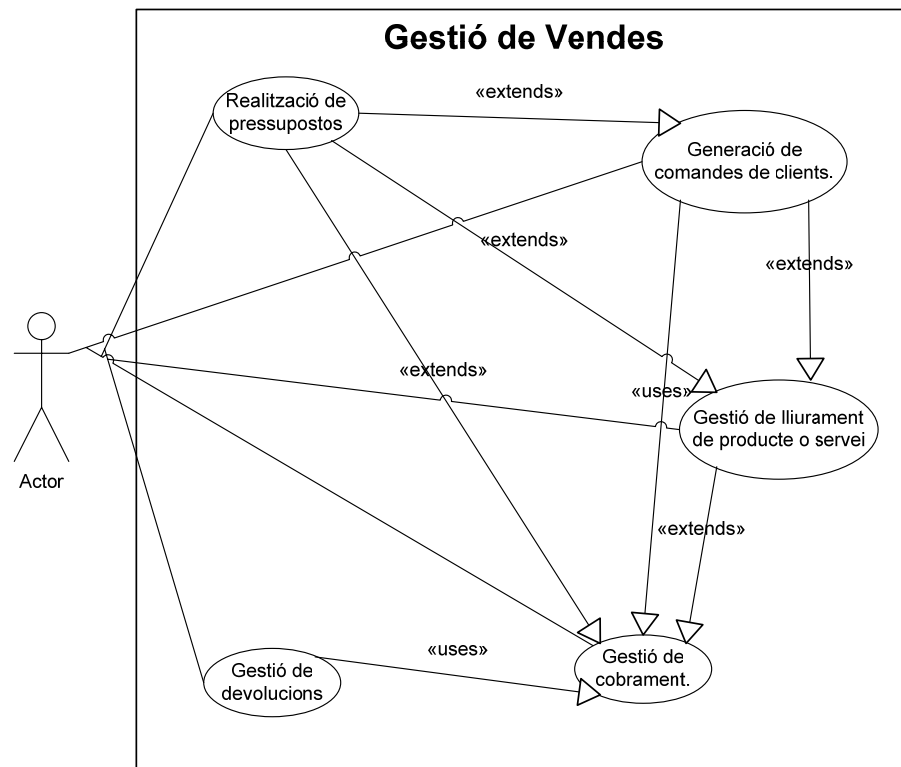


Diagrama de casos d'ús de Gestió de Vendes

## Gestió de compres

### ***Gestió d'entrada de material***

Haurà de permetre fer entrada de noves quantitats d'articles que hi ha en estoc. En el cas d'haver canvis en els preu s'haurà de gestionar a partir de Gestió d'articles.

L'entrada de material també es podrà fer donant per entregades la comada del proveïdor

### ***Gestió de pagaments***

Per tal de tenir totes les despeses entrades al sistema. Caldrà que cada pagament faci referència a un proveïdor, el seu codi de factura que ens ha entrega, la data i el total pagat. En cas de no ser pagada la deixarem en estat de no pagada.



### **Gestió de devolucions**

En el moment que em pagat un proveïdor i després s'ha tornat material. S'haurà de gestionar quin és el total de diners que ens ha retornat, com en el cas de gestió de pagaments, i tot seguit fer el control d'estoc com s'explicarà en el control d'estoc de magatzem.

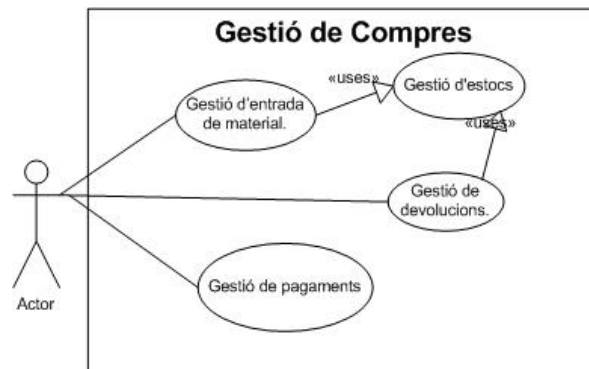


Diagrama de casos d'ús de gestió de compres

### **Gestió de control de magatzem**

#### ***Realització de comandes per enviar a proveïdors.***

Per realitzar determinades comandes s'haurà de fer un document amb el proveïdor i els articles que aquest proveïdor ens serveix. De tal manera que la comanda serà per un proveïdor on s'hi podrà afegir els articles que ell ens serveix.

#### ***Consulta articles amb estoc baix.***

Aquesta consulta ens haurà de treure un llistat de tots els articles en estoc baix i quin és el proveïdor habitual de l'article.

#### ***Gestió estocs.***

Haurà de permetre entrar i retirar estoc d'un article.

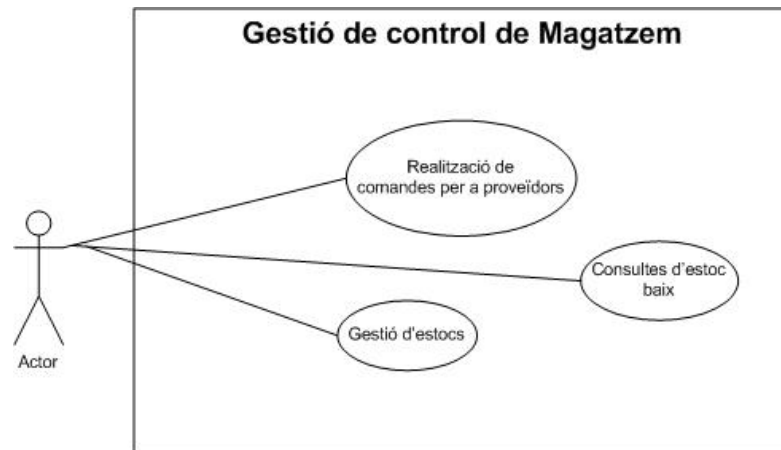


Diagrama de casos d'ús de Gestió de control de magatzem

## Consultes

### *Consulta client*

Donat un client ens mostrà el llistat de factures que se li han entregat, el llistat d'articles que ha comprat des de que va fer la primera compra, el total que se li ha facturat i ha pagat i els deutes que té. Mostrar un gràfic del total facturat.

### *Articles*

Rànquing dels 10 productes més venuts. Llistat dels articles amb estoc, estoc mínim i estoc màxim. Mostrar un gràfic amb el total de productes que s'ha comprat.

### *Proveïdor*

Donat un proveïdor mostrar total de facturació pagada i el llistat de factures en deute, també haurà de mostrar el total que se li ha pagat i el total que se li deu. Mostrar un gràfic de les despeses fetes amb aquest proveïdor.

### *Ingressos i guanys*

Mostrar un gràfic amb els guanys classificats per cada any, més el total ingressat i també mostrar un gràfic de les despeses generades cada any i el total.

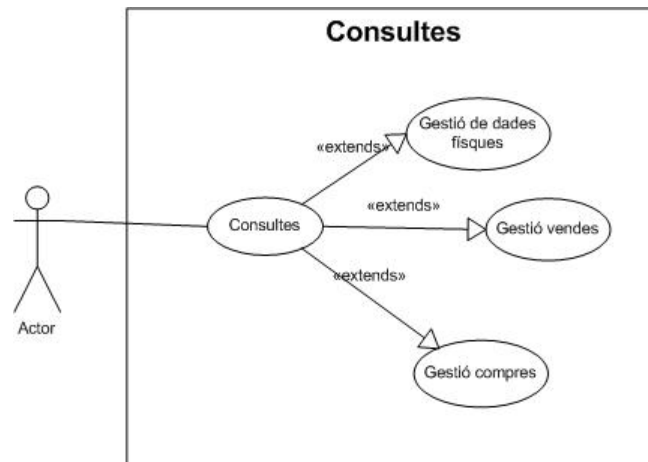


Diagrama de casos d'ús de Consultes

### 3.2.2 Especificació de casos d'us

#### ***Gestió de dades físiques***

#### ***Gestió de l'empresa.***

Aquesta funcionalitat es refereix a que tots els documents oficials a entregar a clients han de tenir uns paràmetres fixes, com són el nom de l'empresa, el codi d'identificació fiscal de l'empresa, l'adreça, codi postal, població, telèfons, correu electrònic, pàgina web, logotip.

#### ***Gestió de client.***

Quan l'usuari doni d'alta un client el sistema li donarà un codi de client automàticament. Les dades necessàries i obligatòries per donar d'alta un client són DNI/NIF, nom, nom comercial, data d'alta, direcció, població, país i telèfon. Les dades opcionals serien un segon número de telèfon, el fax, la direcció de correu electrònic, l'adreça de la pàgina web i un nom de la persona de contacte.

Les dades financeres necessàries però no obligatòries són el compte corrent se'n podrà entrar un, el descompte per pagament immediat, descompte comercial, recàrrec financer, IVA, RE (recàrrec equivalència) i el risc per no pagament que es cregui que es pot tenir amb aquest client. També pot interessar guardar cada quan es pagarà el client i quina es la tarifa que se li aplica, en cas de treballar amb diferents tarifes, de totes maneres si no es treballa amb tarifes la primera (A) és la que s'aplicarà per defecte.

Cada un dels clients podrà estar assignat a una zona i també podrà tenir assignat un treballador que en serà el responsable. En cas necessari es podran posar observacions del client que es creguin necessàries.

Des d'aquest apartat podrem donar d'alta, modifica i eliminar clients, però també fer consultes sobre un client i a part de les dades esmentades anteriorment és podran veure els deu últims contactes que hi ha hagut amb el client, és a dir, l'extracte de les factures emeses.

### **Gestió de proveïdor.**

Com en el cas de manteniment de client alhora de donar d'alta un proveïdor tindrem un seguit de dades obligatòries com: DNI/NIF, nom, nom comercial, data alta, direcció, població, país i telèfon, també hi haurà les dades que poden ser necessàries, però no obligatòries a saber com són un segon número de telèfon, el número de fax, la direcció de correu electrònic, la direcció de la pàgina web de l'empresa i un nom de persona de contacte. En aquest apartat també el programa s'encarregarà de donar un codi al proveïdor.

Les dades financeres, que es coincidiren no obligatòries, són un número de compte corrent, descompte que se'ns aplica per pagament immediat, descompte, el recàrrec financer que se'ns imposa en cas de retard en el pagament, l'iva i RE (Recàrrec d'equivalència), el risc d'impagament que fitxa el proveïdor i la forma de pagament que se li farà, si es cada setmana, quinze dies, mensual o trimestral.

Una de les altres dades que es poden considerar a guardar es el tipus d'atenció que amb la que ens em de dirigir amb el proveïdor i alguna observació que es volgués afegir.

Els últims contactes comercials que hi ha hagut seran visibles en cas de fer una consulta o modificació.

### ***Gestió de treballadors.***

Alhora de donar d'alta, modificar o consultar un treballador s'haurà de poder entra, modificar o veure un seguit de dades obligatòries com seran, el DNI, nom, data alta, direcció, població, país, telèfon, forma de pagament, hores que treballa i el sou. Mentre que les dades no obligades són un segon telèfon, mail i número de compte corrent.

Si l'empresa treballa amb zones a cada un dels treballadors se li podran assignar zones i si existeixen ocupacions de treball també a cada un se li donarà un perfil de treballador, com pe exemple un treballador és administratiu.

Des d'aquest apartat es podrà també donar de baixa al treballador.

### ***Gestió d'articles.***

Tot els serveis i articles que l'empresa ofereix s'hauran de tenir entrats a la base de dades per tal de poder entrar-los als documents i poder fer una gestió dels estocs per tal de poder-ne fer un seguiment tant del producte com de l'empresa.

De cadascun dels article serà necessàries dades com el nom descriptiu, per poder identificar el producte, tot hi que el programa li donar un codi per defecte a cada un dels productes o serveis, l'iva, preu de compra, preu de venda i l'increment. Com a dades addicionals que es podrien incloure serien, imatge, estoc mínim i màxim, unitats/caixes.

A cada article es podrà classificar amb una família o subfamília. Com que un article pot ser servit per un o varis proveïdors per cada un d'ells és podrà guardar el preu de venda del producte, descompte que ens aplica. D'aquets

proveïdor un d'ells serà el proveïdor habitual, tot hi que és podrà canviar de proveïdor habitual.

### ***Gestió de famílies i subfamílies***

Per cada família es guardarà un codi identificatiu que el programa generarà per defecte i el nom. Com a informació no obligatòria a guardar deixarem un cap descriptiu.

Una família podrà ser subfamília d'una altre família i així successivament, es a dir, que hi hauran diferents nivells familiars.

### ***Gestió de zones***

En aquest manteniment s'haurà de poder donar d'alta, modificar, consultar i eliminar zones. La dada necessària serà el nom a part del codi que el programa li assignarà pe defecte, també es podrà donar una descripció de la zona, en cas necessari.

En aquesta pàgina es farà l'assignació de zones a treballadors i a clients, de la mateixa manera que s'ha explicat a manteniment de clients i manteniment de proveïdors, és a dir, un client només se li podrà assignar una zona de manera que una zona podrà tenir molts clients i un treballador podrà tenir moltes zones assignades i una zona podrà tenir molts treballadors travelant-hi.

### ***Gestió d'entitats bancàries***

En el cas que interessi tenir la informació de les entitats bancàries amb les que treballen els client, proveïdor i treballadors, les haurem de poder entrar, consultar, modificar i eliminar si s'escau.

Les dades necessàries sobretot seran el número d'entitat, número d'oficina, per poder-les identificar, el nom de l'entitat i el nom de la oficina, això com a dades obligatòries, i llavors la informació de més a saber serà l'adreça, població, país, telèfons, fax i persona de contacte (director de l'oficina).

## **Gestió de vendes**

### ***Introduir comandes dels clients***

Les comandes faran referència a un client que existirà al sistema i un llistat d'articles que també estaran dins el sistema i el seu estoc serà diferent a zero. De cada article se'n guardarà la quantitat demanada, el preu de venda, el descompte en cas de fer-ne, el preu net de l'article, l'iva i l'import total sense iva. El document tindrà associada, una base imposable, la suma d'iva, la suma de descompte, la suma de recàrrecs i impostos i el total del document.

Cada vegada que entrem un article s'haurà de fer el control d'estoc, ja que, són articles que queden reservats al client.

Aquest document es podrà fer a partir d'un pressupost prèviament realitzat al client i que no hagi estat aprovat. Cal dir que si es fa a partir d'un pressupost aquest un cop generada la comanda, aquest no es podrà utilitzar més i estarà associat a la comanda. Les dades que variaran entre pressupost i la comanda associada seran les dates d'entrega i les de venciment que es podrà modificar.

En cas de voler modificar una comanda s'ha de tenir en compte que si està relacionada amb un pressupost, s'haurà d'eliminar la comanda, modificar el pressupost i llavors tornar a generar la comanda.

Només es podran eliminar comandes que no estiguin servides, es a dir, que cap altre document faci referència a elles. En cas de que una comanda s'elimini i estigui realitzada a partir d'un pressupost aquest passarà a l'estat de no aprovat.

### ***Realitzar pressupostos***

El pressupost faran referència a un client que existirà al sistema i un llistat d'articles que també estaran dins el sistema i el seu estoc serà diferent a zero. De cada article se'n guardarà la quantitat demanada, el preu de venda, el descompte en cas de fer-ne, el preu net de l'article, l'iva i l'import total sense iva. El document tindrà associada, una base imposable, la suma d'iva, la suma de descompte, la suma de recàrrecs i impostos i el total del document.

Cada vegada que entrem un article s'haurà de fer el control d'estoc, ja que, són articles que queden reservats al client.

El pressupost es guardarà com a no acceptat, fins que no sigui utilitzat per un altre document com comanda, factura o albarà.

Per modificar o eliminar un pressupost es podrà fer sempre i quan aquest no estigui acceptat.

### ***Gestionar lliurament d'article.***

Cada vegada que un client rep una remesa de productes o bé algun servei de la nostra empresa, se li haurà de transmetre un albarà on hi constarà el conjunt de productes o serveis, l'import total, si hi ha descomptes, iva...

Alhora de realitzar un albarà es podrà fer directament com en el cas de la realització de pressupostos o introduir comades de clients, sempre i quan aquest documents no estiguin associats a un altre document que els utilitzi.

En el moment de realitzar un albarà a partir d'un pressupost es farà del codi de pressupost, en canvi alhora de fer-ho a partir de comanda es farà de tal manera que de totes les comandes pendents de entre un codi de client i un altre i de entre dues dates es creï un albarà per cada comanda.

Si es vol modificar un albarà es podrà fer sempre i quan aquest no estigui relacionat amb cap comanda o pressupost, en el cas d'estar relacionat s'haurà d'eliminar, la qual cosa el document associat quedarà amb un estat de no utilitzat, de manera que serà possible modificar-lo i, per tant, s'haurà de torna a generar l'albarà com s'ha esmentat anteriorment.

Dit en el paràgraf anterior en el moment d'eliminar un albarà els documents associats quedaran lliures de la relació, per tant, es podran tornar a utilitzar.

### ***Gestió de cobrament***

Caldrà enviar factures al client amb el conjunt de serveis prestats o entregues de productes fetes. les factures es podran realitzar també a partir d'albarans,



comandes o pressupostos. en qualsevol cas, caldrà indicar que els albarans o comandes o pressupostos han estat ja facturats, es a dir, realitzats.

Aquestes factures es podran fer directes com albarans, comandes o pressupostos o a partir dels documents esmentats en anterioritat. Com el cas dels albarans (gestionar lliurament d'article) s'haurà de poder fer de totes les comandes no relacionades amb cap document fer una factura per cada comanda. I en el cas dels albarans per tots els albarans no relacionats amb cap factura doncs agrupar-los per cada client i fer una factura de tots els albarans no facturats.

Les factures no poden ser eliminades, ja que per llei han de tenir un codi correlatiu i podrà ser modificable només si ha estat realitzada directament i el client no l'hagi pagat.

#### **Gestió de devolucions.**

En el cas que el client retorni un producte, cal generar una factura directa amb import negatiu.

***Nota aclarativa:*** per cada pressupost, comanda o albarà entrat en qualsevol dels documents anteriors aquest no s'haurà de poder reutilitzar en el mateix tipus de document. Es a dir que em fet una factura amb varis albarans aquest no es podran reutilitzar en cap altre factura, només es podrà fer en el cas que la factura haguí estat anul·lada.

## ***Gestió de compres***

### ***Gestió d'entrada de material.***

Cada vegada que un proveïdor faci lliurament de material, farà arribar un albarà, a partir d'aquest document es podrà saber quines són les entrades de producte hi ha al magatzem. Per tant necessitarem guardar quina es la nova quantitat de producte que entra a l'empresa.

### ***Gestió de pagaments.***

Cada vegada que fem una compra a un dels proveïdors arribarà una factura que s'haurà d'entrar per tal de poder fer un seguiment de les diferents despeses que s'han produït. caldrà gestionar el pagament d'aquesta factura, guardant el codi de proveïdor, el codi de la factura, la data d'entrega de la factura, el total a pagar i si s'ha pagat.

En cas de ser una factura ja entrada el sistema haurà d'avisar a l'usuari.

### ***Gestió de devolucions***

En l'entrega de material per par del proveïdor pot ser que s'hagi equivocat amb el producte que s'havia demanat, aquest arribi amb mal estat, hi hagi més quantitat de la demanada, ...

Per tant farem una gestió de pagaments amb negatiu i tot seguit es passarà a una gestió de sortida de material (s'explica més endavant).

## ***Control d'estoc de magatzem***

### ***Realització de comandes per enviar a un proveïdor***

La comanda faran referència a un proveïdor que existirà al sistema i un llistat d'articles que també estaran dins el sistema i estaran relacionats amb el proveïdor.

En el moment d'afegir articles a la comanda per defecte sortirà el resultat de la

resta entre l'estoc existent i l'estoc màxim que hi pot haver.

De cada article se'n guardarà la quantitat demanada, el preu de venda, el descompte en cas que el proveïdor en faci, el preu net de l'article, l'iva i l'import total sense iva. El document tindrà associada, una base imposable, la suma d'iva, la suma de descompte, la suma de recàrrecs i impostos i el total del document.

La comanda es guardarà com a no realitzada.

Quan arribi la comanda provinent de proveïdor i estigui correcte podrem donar-la per realitzada i per tant automàticament el sistema passarà a fer el recompte d'estoc.

Per modificar o eliminar una comanda aquesta haurà d'estar en esta no realitzada.

### ***Consulta articles amb estoc baix.***

Per tal que l'usuari sàpiga si ha de fer comanda en algun proveïdor en aquesta funcionalitat li sortiran tots els productes que estan amb estoc per sota o igual al mínim i indicat quin és el proveïdor habitual per tal que sigui més senzill al moment de saber a quin proveïdor fer la comanda.

### ***Afegir o retirar estocs***

En el cas que d'haver de fer entrada de material o retirada degut a circumstancies que no són per haver servit a un client o haver arribat una comanda d'un proveïdor s'haurà de poder fer la gestió tant d'entrada com de sortida d'articles, per tal de tenir actualitzat l'estoc en el moment de realitzar qualsevol document o consulta, ja que no estaria quadrat el que hi ha en realitat al magatzem al que indica el programa.

Per tant s'haurà de guardar quina és la quantitat que tenim de cada article en el magatzem.

### 3.3 Anàlisi dels casos d'ús

Duran l'anàlisi dels casos d'ús cal identificar les classes associades a cada un d'ells i la interacció entre els diferents objectes.

#### ***Gestió de dades físiques.***

##### ***Gestió de l'empresa.***

- Empresa: Ha de permetre gestionar la informació de l'empresa com el seu nom de l'empresa

##### ***Gestió de client.***

- Client: Ha de permetre gestionar tota la informació referent a cada client des de les dades referents a l'empresa (DNI/NIF, nom, nom comercial, adreça, població, país, telèfons de contacte, fax, persona de contacte a l'empres, pàgina web i adreça electrònica), dades bancàries, dades econòmiques com els descomptes comercials, per pagament immediat, recàrrec financer, iva, re, forma de pagament, tarifa que s'aplica.
- Zona: classe que identifica la zona amb les que l'empresa pot dividir el territori pel que es mou.
- Treballador: Classe que guarda totes les dades referents al treballador de l'empresa des de les seves personals, com el nom, DNI, direcció del domicili, població on viu, telèfons de contacte i adreça electrònica. Les dades bancàries, el sou que cobra, les hores que treballa, la forma de pagament i la ocupació que té a l'empresa.

**Gestió de proveïdor.**

- Proveïdor: la informació que ha de poder gestionar són les dades referents a l'empresa (DNI/NIF, nom, nom comercial, adreça, població, país, telèfons de contacte, fax, persona de contacte a l'empres, pàgina web i adreça electrònica), dades bancàries, dades econòmiques com els descomptes comercials, per pagament immediat, recàrrec financer, iva, re, forma de pagament, el tipus d'atenció que s'ha de tenir en compte al hora de contactar-hi.

**Gestió de treballadors.**

- Treballador: explicat en Gestió de clients
- Zona: explicat en Gestió de clients.
- AssociacióTreballadorZona: Classe que encapsula quins treballadors treballen a cada zona.

**Gestió d'articles.**

- Article: Classe encarregada de guardar informació referent al productes o serveis que ofereix l'empresa al clients. Les dades a guardar són un nom que descrigui l'article, una descripció de les característiques, la imatge, l'iva que se l'hi ha d'aplicar, quantes unitats hi ha a cada caixa de venta, l'estoc mínim, l'estoc màxim, l'estoc total que hi ha, el cost segons el proveïdor, el cost net segons el proveïdor, el marge aplicat en les diferents tarifes (quatre en total), el preu per cada una de les tarifes, el descompte segons el proveïdor.
- Família: Classe que gestiona les diferents famílies amb les que es pot classificar un article. Caldrà guardar el nom de la família i una descripció informativa.
- Subfamília: De la mateixa maner que la família gestiona les diferents famílies en que es pot classificar un article, aquesta gestiona amb quines subfamílies es poden subclassificar els articles.
- AssociacióArticleProveïdor: Classe que encapsula per cada article els diferents proveïdors que proveeixen l'empresa d'aquest article. En guardarà

el cost del producte, el descompte que el proveïdor ofereix i el cost net total de la compra de l'article.

- Proveïdor: Classe ja explicada en gestió de proveïdors.

### ***Gestió de famílies i subfamílies***

- Família: Classe explicada en gestió d'article.
- Subfamília: Classe explicada en gestió d'article.

### ***Gestió de zones***

- Treballador: explicat en Gestió de clients.
- Zona: explicat en Gestió de clients.
- AssociacióTreballadorZona: explicat en gestió de Treballadors.
- Client: explicada en gestió d'articles.

### ***Gestió d'entitats bancàries***

- EntitatBacaria: classe que conté la informació de les diferents entitats bancàries i oficines que tenen els diferents clients, proveïdors i treballadors de l'empresa.

### ***Gestió de vendes***

#### ***Realitzar pressupostos***

- PressupostVenta: Classe que encapsula el document de demanda d'articles que ha fet un client. S'haurà de guardar el número de seria del document, el número de document, la data d'entrega, la data de venciment, els valor dels descomptes i impostos, els subtotals (bases imposable, suma dels imports) i el total.
- LiniaPressupost: Classe que guarda de cada article que forma part del pressupost la quantitat, el preu que s'ha venut, el descompte aplicat a l'article, preu net total i total de la línia

- Article: classe explicada a gestió d'articles.
- Client: classe explicada a gestió de clients.

### ***Introduir comandes dels clients***

- ComandaVenta: Classe que encapsula el document de demanda d'articles que ha fet un client. S'haurà de guardar el número de seria del document, el número de document, la data d'entrega, la data de venciment, els valor dels descomptes i impostos, els subtotals (bases imposable, suma dels imports) i el total.
- LiniaComanda: Classe que guarda de cada article que forma part de la comanda la quantitat, el preu que s'ha venut, el descompte aplicat a l'article, preu net total i total de la línia
- Article: classe explicada a gestió d'articles.
- Client: classe explicada a gestió de clients.
- PressupostVenta: classe explicada a realitzar pressupostos.
- LiniaPressupost: classe explicada a realitzar pressupostos.

### ***Gestionar lliurament d'article.***

- AlbaraVenta: Classe que encapsula el document de lliurament d'articles que ha fet un client. S'haurà de guardar el número de seria del document, el número de document, la data d'entrega, la data de venciment, els valor dels descomptes i impostos, els subtotals (bases imposable, suma dels imports) i el total.
- LiniaAlbara: Classe que guarda de cada article que forma part de l'albarà la quantitat, el preu que s'ha venut, el descompte aplicat a l'article, preu net total i total de la línia
- Article: classe explicada a gestió d'articles.
- Client: classe explicada a gestió de clients.
- PressupostVenta: classe explicada a realitzar pressupostos.
- LiniaPressupost: classe explicada a realitzar pressupostos.
- ComandaVenta: classe explicada a introduir comandes de client.
- LiniaPressupost: classe explicada a introduir comandes de client.

**Gestió de cobrament**

- FacturaVenda: Classe que encapsula el document que fa referència als articles que el client haurà de pagar per haver rebut la comanda. S'haurà de guardar el número de seria del document, el número de document, la data d'entrega, la data de venciment, els valor dels descomptes i impostos, els subtotals (bases imposable, suma dels imports) i el total.
- LiniaFactura: Classe que guarda de cada article que forma part de la factura la quantitat, el preu que s'ha venut, el descompte aplicat a l'article, preu net total i total de la línia
- AlbaraVenda: explicat a gestió de lliurament d'article.
- LiniaAlbara: explicat a gestió de lliurament d'article.
- Article: classe explicada a gestió d'articles.
- Client: classe explicada a gestió de clients.
- PressupostVenda: classe explicada a realitzar pressupostos.
- LiniaPressupost: classe explicada a realitzar pressupostos.
- ComandaVenda: classe explicada a introduir comandes de client.
- LiniaPressupost: classe explicada a introduir comandes de client.

**Gestió de devolucions.**

Utilitza les mateixes classes que gestió de cobrament.

**Gestió de compres****Gestió d'entrada de material.**

- ComandaCompra: Classe que encapsula el document de demanda d'articles que s'ha de fer a un proveïdor. S'haurà de guardar el número de seria del document, el número de document, la data d'entrega, la data de venciment, els valor dels descomptes i impostos, els subtotals (bases imposable, suma dels imports) i el total.
- LiniaComanda: Classe que guarda de cada article que forma part de la comanda la quantitat, el preu que s'ha venut, el descompte aplicat a l'article, preu net total i total de la línia



- Article: explicat a gestió d'articles.

***Gestió de pagaments.***

- FacturaCompra: Classe que encapsula el document que el proveïdor ens entrega després d'haver entregat la comanda. S'haurà de guardar el total a pagar, el número que porta la factura del proveïdor, la data d'Entrega de la factura i la data de Venciment.

***Gestió de devolucions***

- FacturaCompra: explicat a gestió de pagaments.
- Article: explicat a gestió d'articles.

***Control d'estoc de magatzem******Realització de comandes per enviar a un proveïdor***

- ComandaCompra: explicat a gestió d'entrada de material.
- LiniaComanda: explicat a gestió d'entrada de material.
- Article: explicat a gestió d'articles.
- Proveïdor: explicat a gestió de proveïdors.

***Consulta articles amb estoc baix.***

- Article: explicat a gestió d'articles.
- Proveïdor: explicat a gestió de proveïdors.

***Afegir o retirar estocs***

- Article: explicat a gestió d'articles.

### 3.4 Anàlisi de classes

#### *Gestió de dades físiques.*

Per a cada cas d'ús, es passarà a veure quines són les classes implicades i anàlisi de cadascuna d'elles, incloent les dades i operacions. En aquest apartat es veurà l'herència i l'encapsulament de les relacions M..N de les classes.

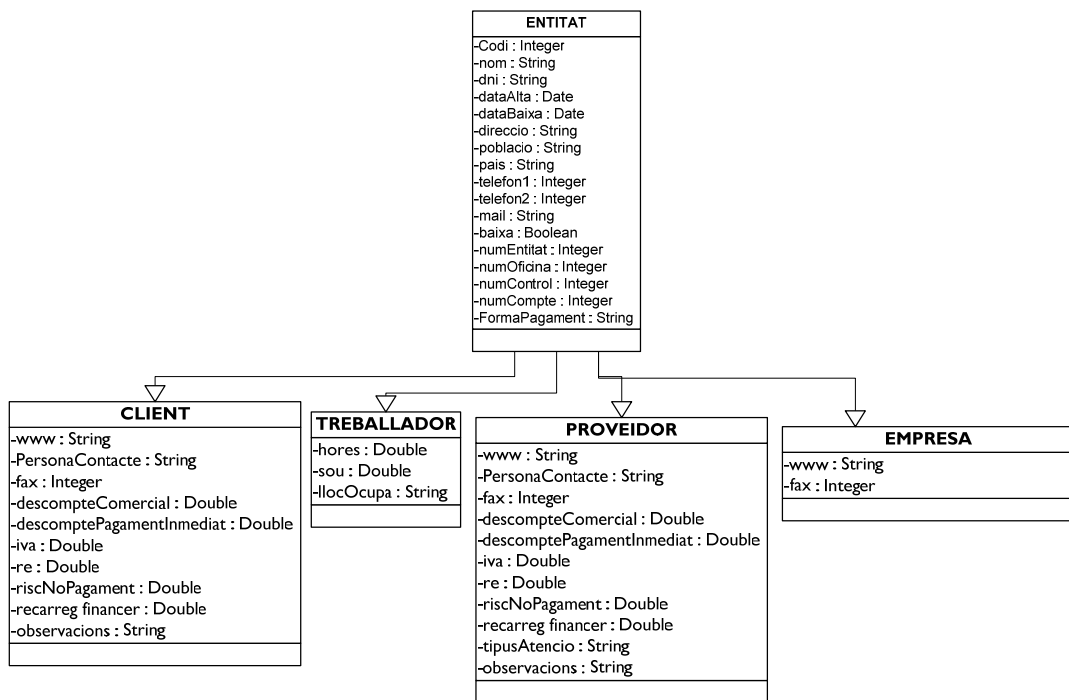
Abans de passar a veure cada un dels casos d'ús, es passarà a fer un estudi de possible herència entre un conjunt de classes, ja que, en apartats anteriors s'ha vist que uns seguit .

#### *Herència entre Treballador, Client i Proveïdor*

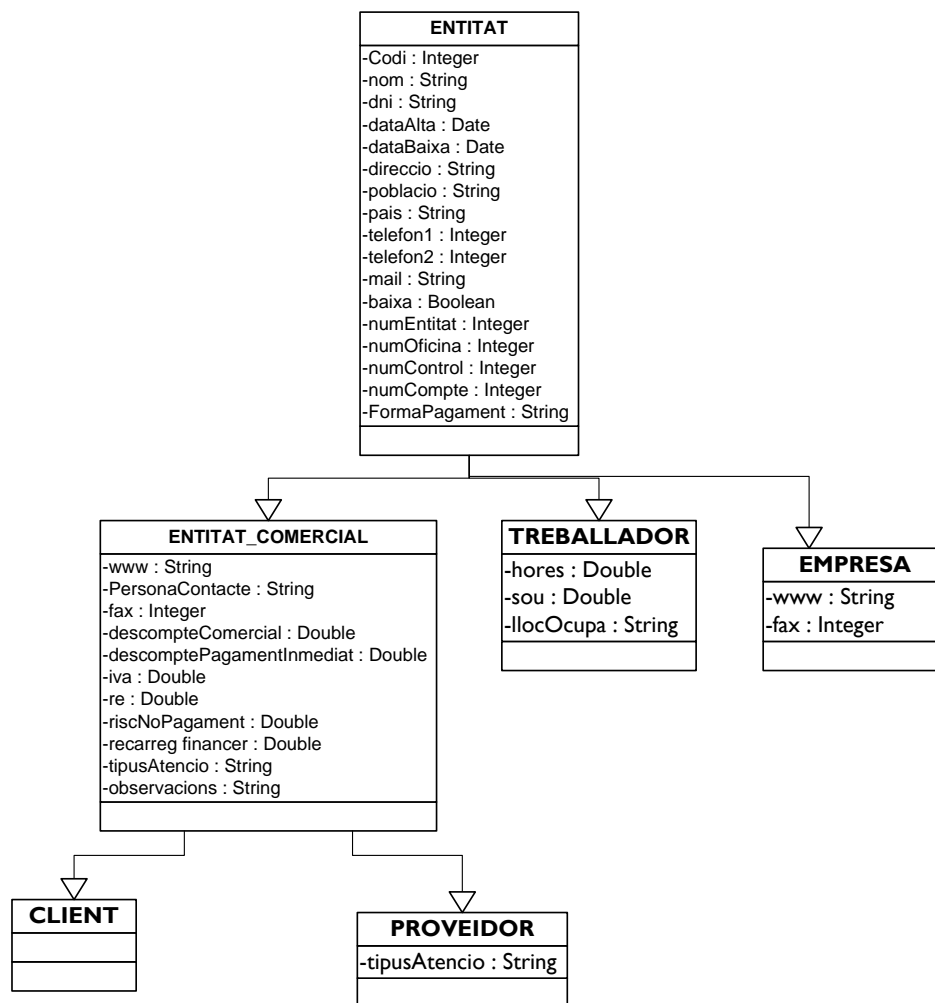
Donat a que treballador, client, proveïdor i l'empresa fan referència a persones o empreses físiques, molts dels atributs seran comuns. Tot seguit es mostren les tres classes amb els atributs que cal saber de cada una:

CLIENT	TREBALLADOR	PROVEIDOR	EMPRESA
-Codi : Integer -nom : String -dni : String -dataAlta : Date -dataBaixa : Date -direccio : String -poblacio : String -pais : String -telefon1 : Integer -telefon2 : Integer -fax : Integer -mail : String -baixa : Boolean -numEntitat : Integer -numOficina : Integer -numControl : Integer -numCompte : Integer -FormaPagament : String -www : String -PersonaContacte : String -fax : Integer -descompteComercial : Double -descomptePagamentInmediat : Double -iva : Double -re : Double -riscNoPagament : Double -recarreg financer : Double -observacions : String	-Codi : Integer -nom : String -dni : String -dataAlta : Date -dataBaixa : Date -direccio : String -poblacio : String -pais : String -telefon1 : Integer -telefon2 : Integer -fax : Integer -mail : String -baixa : Boolean -numEntitat : Integer -numOficina : Integer -numControl : Integer -numCompte : Integer -FormaPagament : String -hores : Double -sou : Double -llocOcupa : String	-Codi : Integer -nom : String -dni : String -dataAlta : Date -dataBaixa : Date -direccio : String -poblacio : String -pais : String -telefon1 : Integer -telefon2 : Integer -fax : Integer -mail : String -baixa : Boolean -numEntitat : Integer -numOficina : Integer -numControl : Integer -numCompte : Integer -FormaPagament : String -www : String -PersonaContacte : String -fax : Integer -descompteComercial : Double -descomptePagamentInmediat : Double -iva : Double -re : Double -riscNoPagament : Double -recarreg financer : Double -observacions : String -tipusAtencio : String	-Codi : Integer -nom : String -dni : String -dataAlta : Date -dataBaixa : Date -direccio : String -poblacio : String -pais : String -telefon1 : Integer -telefon2 : Integer -fax : Integer -mail : String -baixa : Boolean -numEntitat : Integer -numOficina : Integer -numControl : Integer -numCompte : Integer -www : String

Vista la figura anterior podem fer una primera extracció d'atributs, els que fan referència a dades personals. La nova classe s'anomenarà Entitat de la qual se'n derivaran les classes Client, Proveïdor, treballador i empresa



Els objectes fills heretaran les funcions del pare i si cal les reescriuran per adaptar-les a la seva representació. Tot hi haver aplicat herència entre aquestes tres classes, se'n pot fer una altre, ja que, entre client, proveïdor i queden encara atributs iguals.



Finalment el model resultat és el de la figura anterior on d'Entitat surten dues classes que hereten d'ella, Entitat\_Comercial, Empresa i Treballador, on Entitat Comercial representaria per dir d'alguna manera a una empresa, tant és si el client, proveïdor o empresa i Treballador a la persona física que treballa per l'empresa que representa el programa. I per acabar client i proveïdor que hereten d'entitat\_Comercial, que necessitarem diferenciar-ho per alguns atributs i també per que les funcions que realitzaran seran diferent.

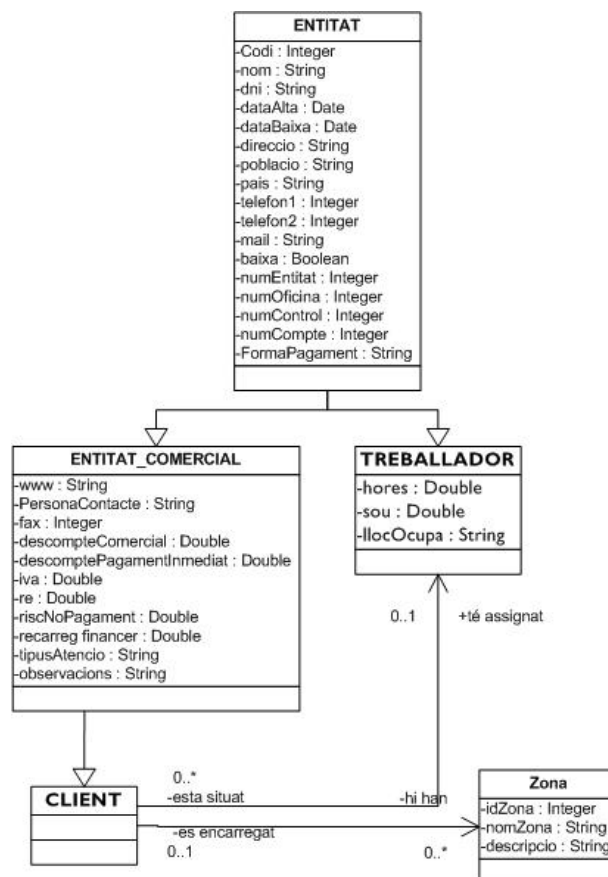
### **Gestió de l'empresa.**

En el manteniment de l'empresa la classe implicada és Empresa, en les quatre operacions que és poden realitzar (alta, modificació i consulta). Cal dir que per

conseqüència d'utilitzar la classe Empresa també intervenen Entitat i Entitat\_Comercial.

Les dades són les que s'han anat anomenant repetidament en els diferents apartats anteriors, les operacions que és podran realitzar en el manteniment seran donar d'alta l'empresa, de manera que només es podrà fer una única vegada, fer modificacions i consultes. L'operació donar de baixa, com en els apartats següents no té sentit.

### Gestió de clients.



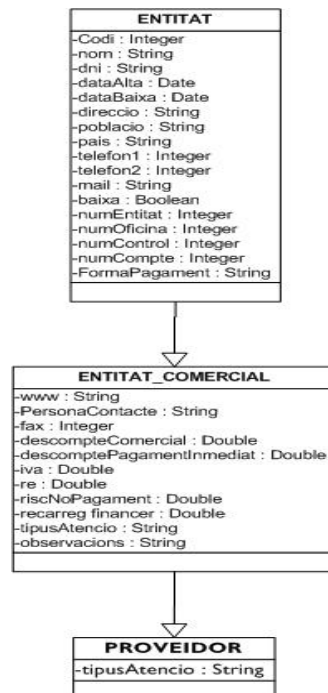
Les operacions necessàries en aquest manteniment a part de les importants com alta, baixa, modificació i consulta, també necessitarem una operació per assignar una zona al client i assignar un treballador, això voldrà dir que necessitarem un llistat de treballadors i una altre de zones per tal de poder conèixer quins tenim donats d'alta en el sistema.

La classe principal que en aquest cas es client tindrà com a clau principal un codi de client i com a claus foranes: codi d'entitat, codi entitat\_comercial, codi de treballador i codi zona, aquestes dues ultimes són a causa de l'associació que hi ha amb les classes treballador i zona.

La integritat referencial alhora de realitzar les diferents operacions. En l'alta d'un client no caldrà que hi hagi cap element creat dels que intervenen, en canvi en les altres operacions no serà així. Modificar, consultar i baixa un client si que client haurà d'existir en el sistema, cal tenir en compte que alhora de donar de baixa un client treballador i zona no quedaran afectats. Finalment les dues operacions d'assignació, per tal d'assignar un treballador o una zona aquestes hauran d'existir en el sistema i en el cas del treballador no podrà estar donat de baixa.

Com que el programa a de ser el màxim de flexible pot ser no necessària l'assignació d'un treballador i d'una zona. Client coneixerà quin treballador el servirà i a quina zona esta assignat.

### **Gestió de proveïdor.**

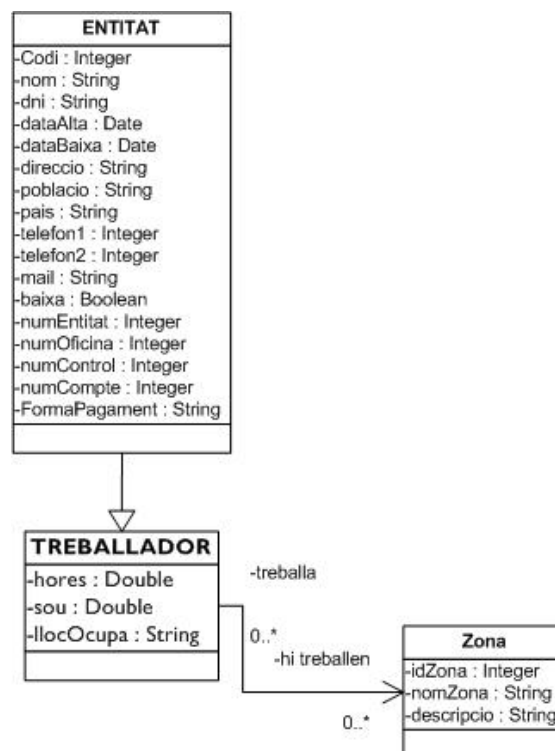


Les operacions necessàries en el manteniment són l'alta, la baixa, la modificació i la consulta on només hi intervenen les classes proveïdor i per consegüent entitat i entitat\_comercial.

Com en el cas anterior i els que en anirem trobat per donar d'alta un proveïdor no farà falta que existeixi al sistema, però si quan fem una consulta, modificació i eliminació.

En aquesta gestió no tenim cap cas de possibles zeros en associacions i explorabilitat a les associacions.

### **Gestió de treballadors.**

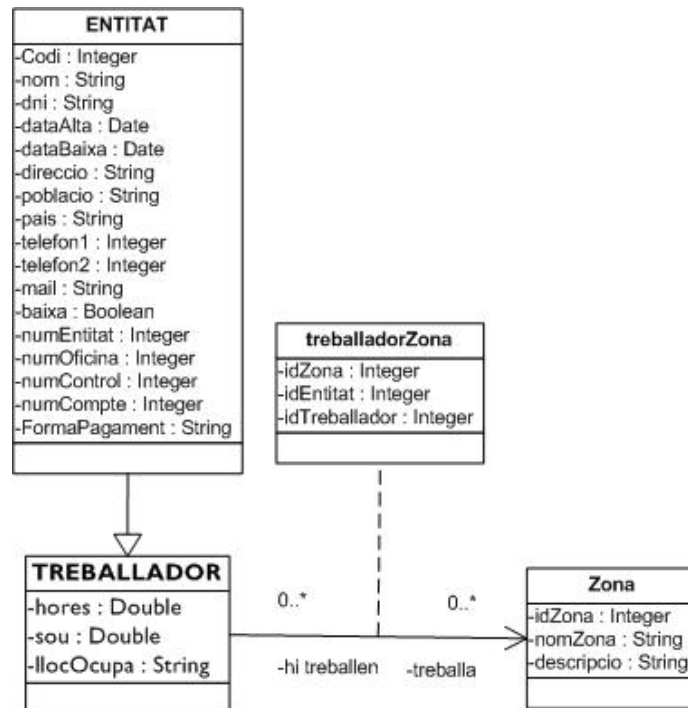


En aquest manteniment les classe que intervenen són treballador i zona, per conseqüència d'intervenir treballador també intervé Entitat.

Les operacions seran alta, baixa, modificació, consulta i assignar Zona. Com en les altres per donar d'alta no es necessari que existeixi cap entitat, ni el treballador, en canvi per modificar, consultar, eliminar, assignar Zona treballador i desassignar Zona al treballador haurà d'existir, per consegüent l'entitat també i llavors per afegir la zona al treballador, aquesta haurà també d'existir.

Pot ser que un treballador no estigui assignat a cap zona, ja que pot ser que l'empresa no treballi per zones, de manera que un treballador coneixerà les zones on treballa.

Per acabar l'anàlisi es pot observar que entre treballador i zona hi ha una associació M..N per tant d'aquesta associació apareix una classe més, així que passarem a tenir el següent diagrama classes per aquest cas d'ús.

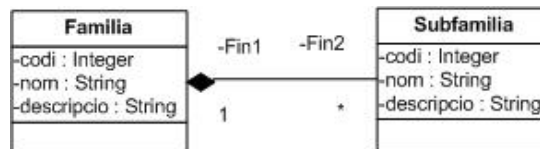


Caldrà tenir en compte que si volem eliminar un treballador també s'haurà d'eliminar les associacions amb zona, per tant serà igual a fer desassignacions.



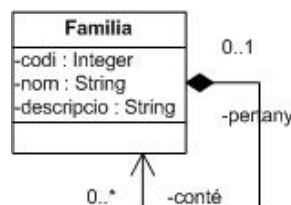
### **Gestió de famílies i subfamílies**

Segons l'especificació feta per el client els productes o serveis s'hauran de poder classificar i subclassificar per això una forma de fer-ho és amb famílies i tot seguit per subclassificació en subfamílies així que el resultat de l'abstracció en classes seria:



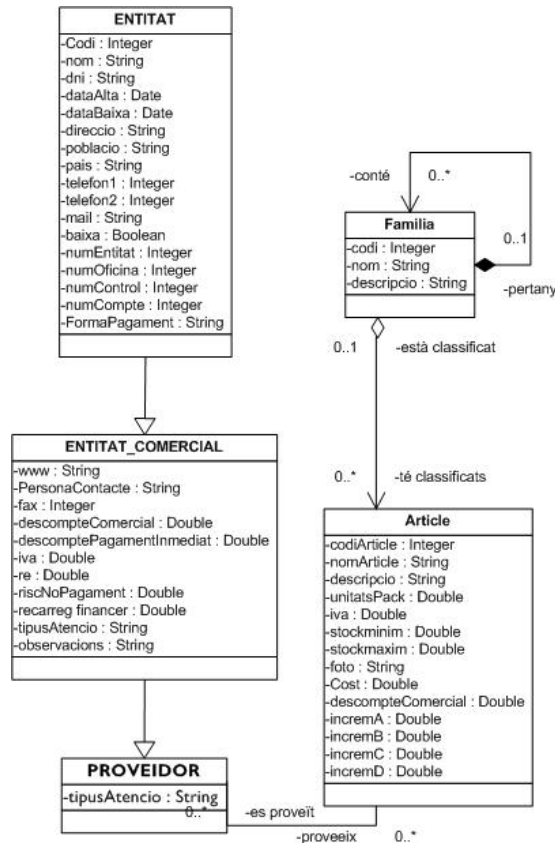
Cal tenir en compte que una subfamília sempre caldrà que estigui associada a una família, de manera que si una família s'elimina totes les subclassificacions també desapareixeran.

Aquesta primera anàlisi no permet que una subfamília sigui també família, es a dir, que només permet fins a dos nivells de classificació, per tant, si família i subfamília són una mateixa classe que estigui associada a ella mateixa, podem fer varis nivells de subclassificació. De manera que quedarà de la següent manera:



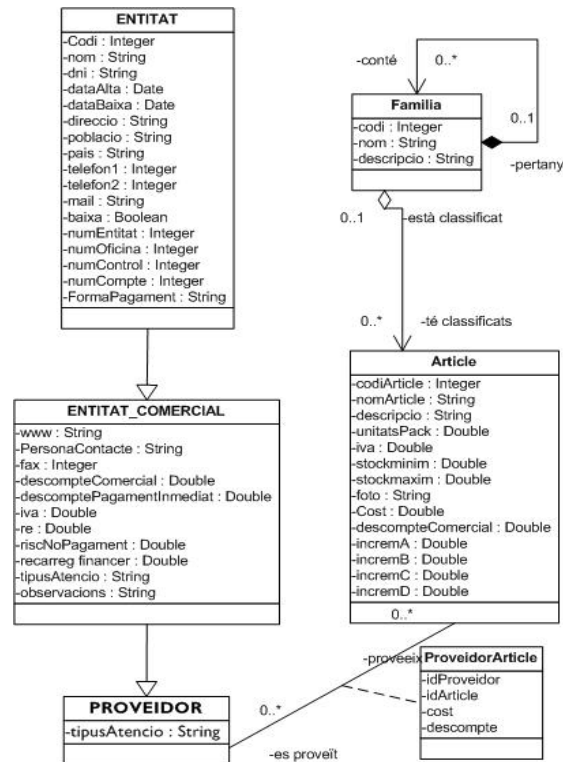
### **Gestió d'articles.**

Com tots els manteniments tindrem les operacions alta, baixa, modificació i consulta, però a més en aquest cas tindrem assignar proveïdor habitual, assignar proveïdor, eliminar proveïdor, assignar família, desassignar proveïdors i desassignar família.



Per donar d'alta un article no caldrà l'existència de les altres, però si en el cas de la modificació, consulta i eliminar un article si que haurà d'existir en el sistema. En el cas de fer les assignacions pertinents els objectes haurà de ser existents en el sistema. En cas d'eliminar un article la família no quedarà afectada, però si les associacions amb proveïdor. És a dir, que si eliminem un article s'hauran d'eliminar totes les associacions amb els proveïdor. En aquest cas veiem que en l'apartat de manteniment de proveïdor on no em tingut en compte article si ho haurem de tenir present, ja que si eliminem un proveïdor totes les associacions amb l'article també s'hauran d'eliminar. També en el cas d'eliminar una família no s'ha esmentat que passa amb l'article de tal manera que en aquest cas no es deixarà eliminar una família si hi ha articles classificats en ella.

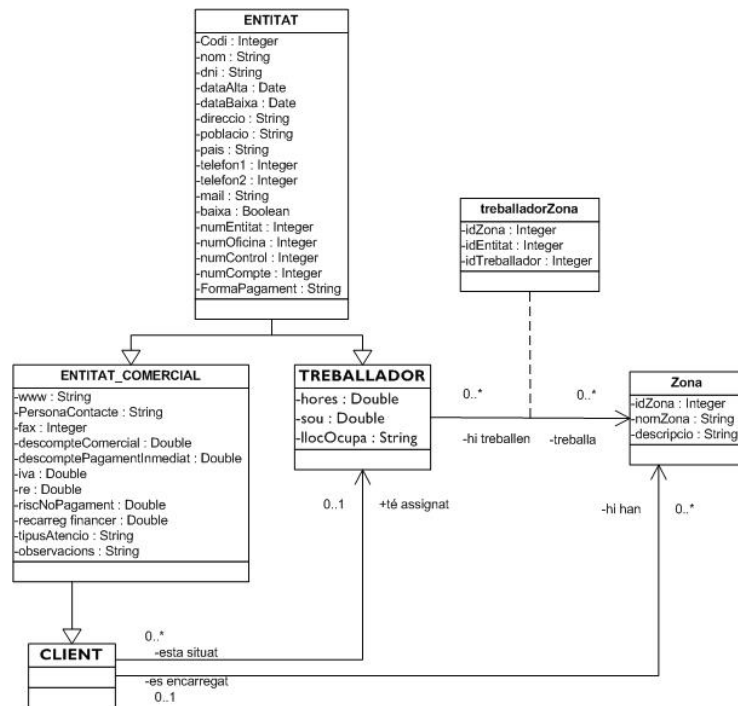
Un cop vist que entre proveïdor i Article existeix una associació M..N el diagrama d'aquest cas d'ús passarà a ser el següent:



On per a cada associació entre proveïdor i article es guardaran els atributs de cost i descompte aplicat.

### Gestió de zones

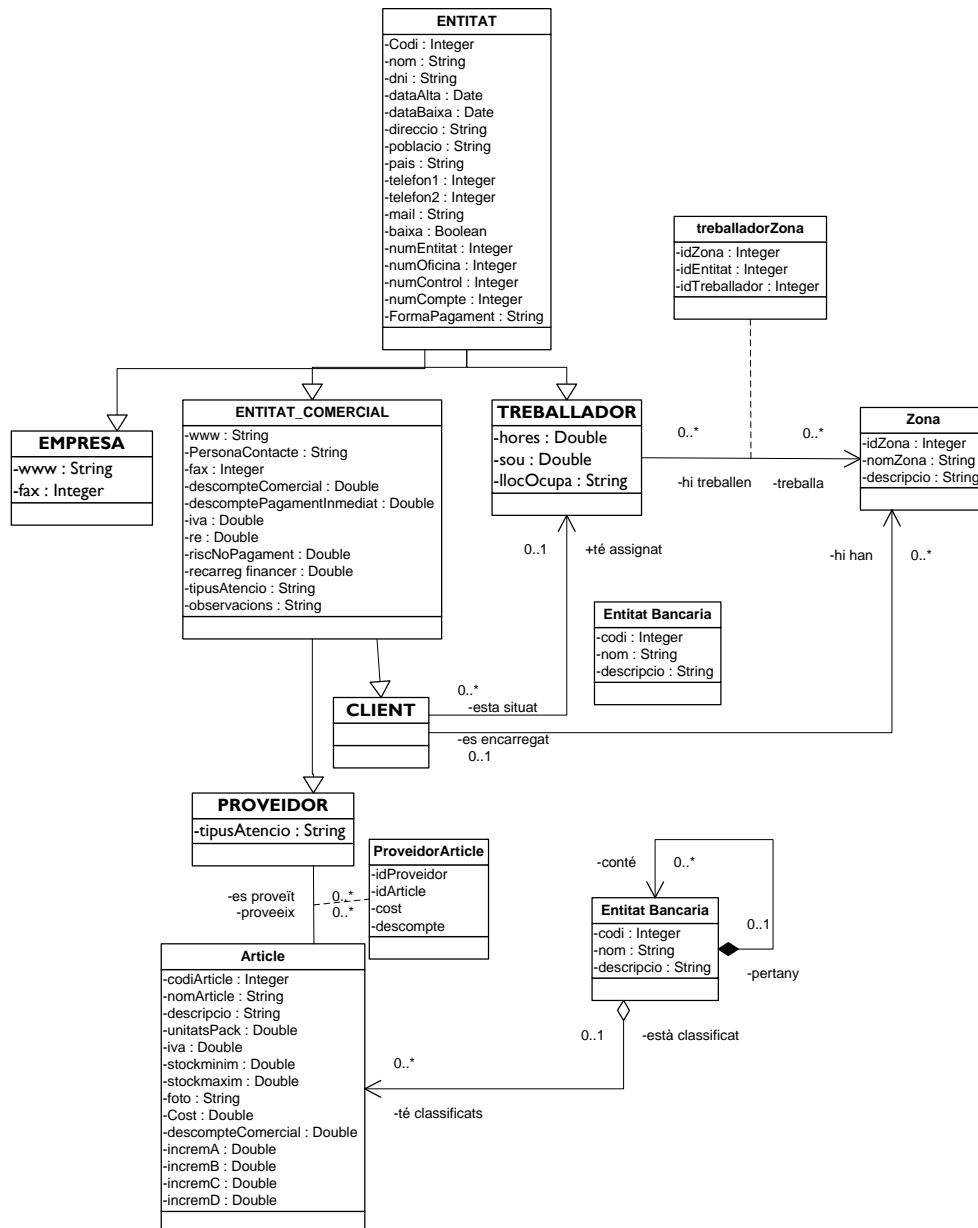
En el manteniment de zones es faran les operacions d'alta, baixa, modificació i consulta, però a més es farà l'associació i desassociació entre treballador i zona i també l'associació i desassociació entre client i zona, per tal de tenir una manera l'usuari tingui de manera més visual aquestes assignacions.



Per donar d'alta una zona no caldrà que existeixi aquesta en el sistema i en el cas de modificar, consulta, eliminar, assignar i desassignar si que la zona haurà d'existir. En el cas d'assignacions de treballador i clients aquest hauran d'existir en el sistema i també en el cas de dessassignar-los. Caldrà tenir en compte que per un client només podrà estar assignat a una zona mentre que aquesta zona podrà tenir molts clients, en el cas de treballadors aquests podran estar assignats a moltes zones i una zona hi treballaran molts treballadors. Com ja s'ha esmentat en la gestió de Treballador.

### Gestió d'entitats bancàries

Com que aquest manteniment es purament informatiu només ens farà falta la classe **Entitat\_Bancaria** per fer les altes, baixes, modificacions i consultes. On cap de les operacions tindrà efectes sobre cap de les classes i manteniments esmentats anteriorment, per la seva única utilitat de ser informativa.

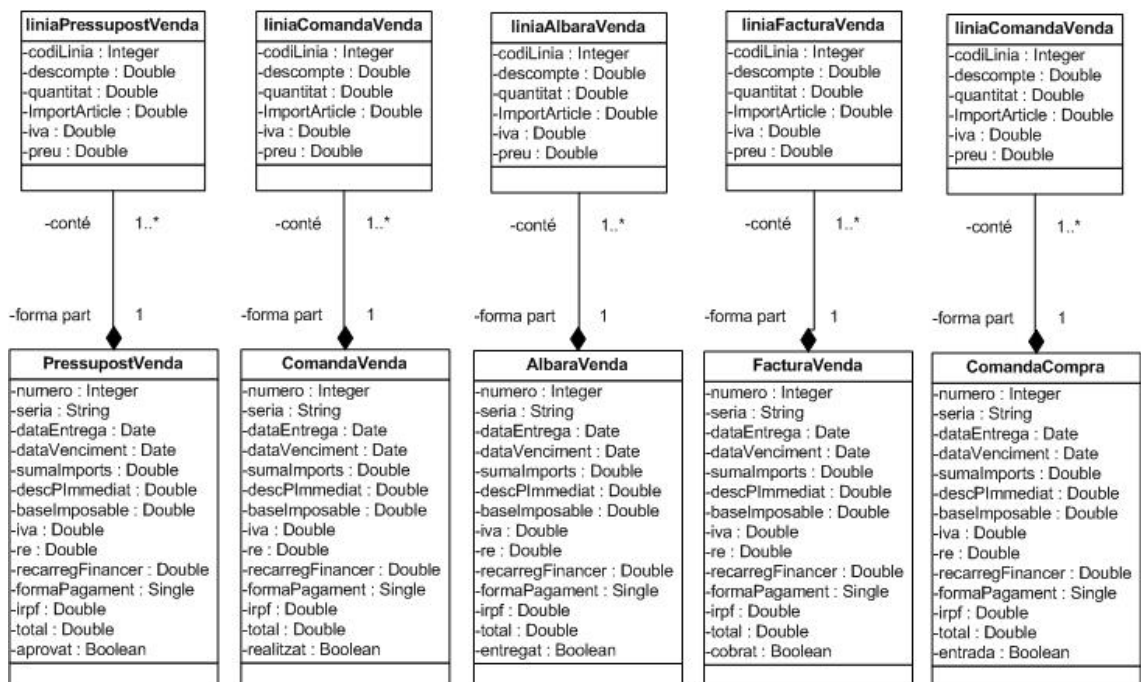
**Diagrama resultant de la gestió de dades físiques****Gestió de vendes**

Abans de començar a veure cada cas d'ús, en el diagrama del model conceptual presentat en anterioritat, hi ha un conjunt de classes que formen part d'aquest subsistema que se'n pot fer un anàlisi de possible herència.

### Herència entre documents

Les classes PressupostVenda, ComandaVenda, AlbaraVenda i FacturaVenda que representen els document que fan referència al client i ComandaCompra que representa els documents que són per el proveïdor, totes elles tenen atributs comuns.

Per conseqüència les classes lligades a elles com són les línies corresponents a cada una de les classes també tenen els atributs comuns.



Aquí podem veure la semblança de totes les classes entre elles, l'únic atribut que els diferencia és un atribut de tipus booleà, que al fi i al cap, representa l'estat del document.

Dit això podem arribar a la conclusió que amb una sola classe que representi totes les classes referents a documents en tenim prou, però si observem bé el diagrama conceptual del model veiem que les relacions entre elles i amb d'altres classes del model són diferents, és per això que no servirà una sola classe que representi totes les classes.

Per començar, primer ens fixarem amb les relacions amb d'altres classes del model que no sigui entre elles. El que s'observa és:

- Totes elles estan relacionades amb la classe Article.
- PressupostVenda, ComandaVenda, AlbaràVenda i FacturaVenda estan relacionades amb la classe Client
- ComandaCompra està relacionada amb la classe Proveïdor

Tot seguit les relacions entre elles:

- PressupostVenda: ComandaVenda.
- ComandaVenda: PressupostVenda i AlbaraVenda.
- AlbaraVenda: ComandaVenda, PressupostVenda i FacturaVenda.
- FacturaVenda: AlbaraVenda, ComandaVenda i PressupostVenda

Fetes aquestes observacions podem arribar a la conclusió següent:

- Utilitzar una classe que representi a totes les classes i passarà estar relacionada amb Entitat (classe explicada en el cas d'ús de Gestió de dades físiques)→ classe Document.
- ComandaVenda, AlbaraVenda i PressupostVenda són tres classes que poden ser utilitzades entre elles o per FacturaVenda. Per tant les podem englobar en una subclasse de Document, anomenada DocumentClient
- FacturaVenda serà una subclasse de Document, ja que com s'ha exposat anteriorment DocumentClient engloba els documents que poden ser utilitzats per d'altres document i el cas de la FacturaVenda no ho podrà ser.
- ComandaCompra serà una subclasse de Document, ja que està relacionada amb proveïdor.

Per acabar les classes que representen les línies dels documents passarà a ser una única classe, ja que totes tenen els atributs comuns un a un, estan relacionades amb la classe article i relacionades amb tots els tipus de documents, per tant, aquesta estarà relacionada amb la classe Document.

Tot seguit es mostra el resultat de l'anàlisi:

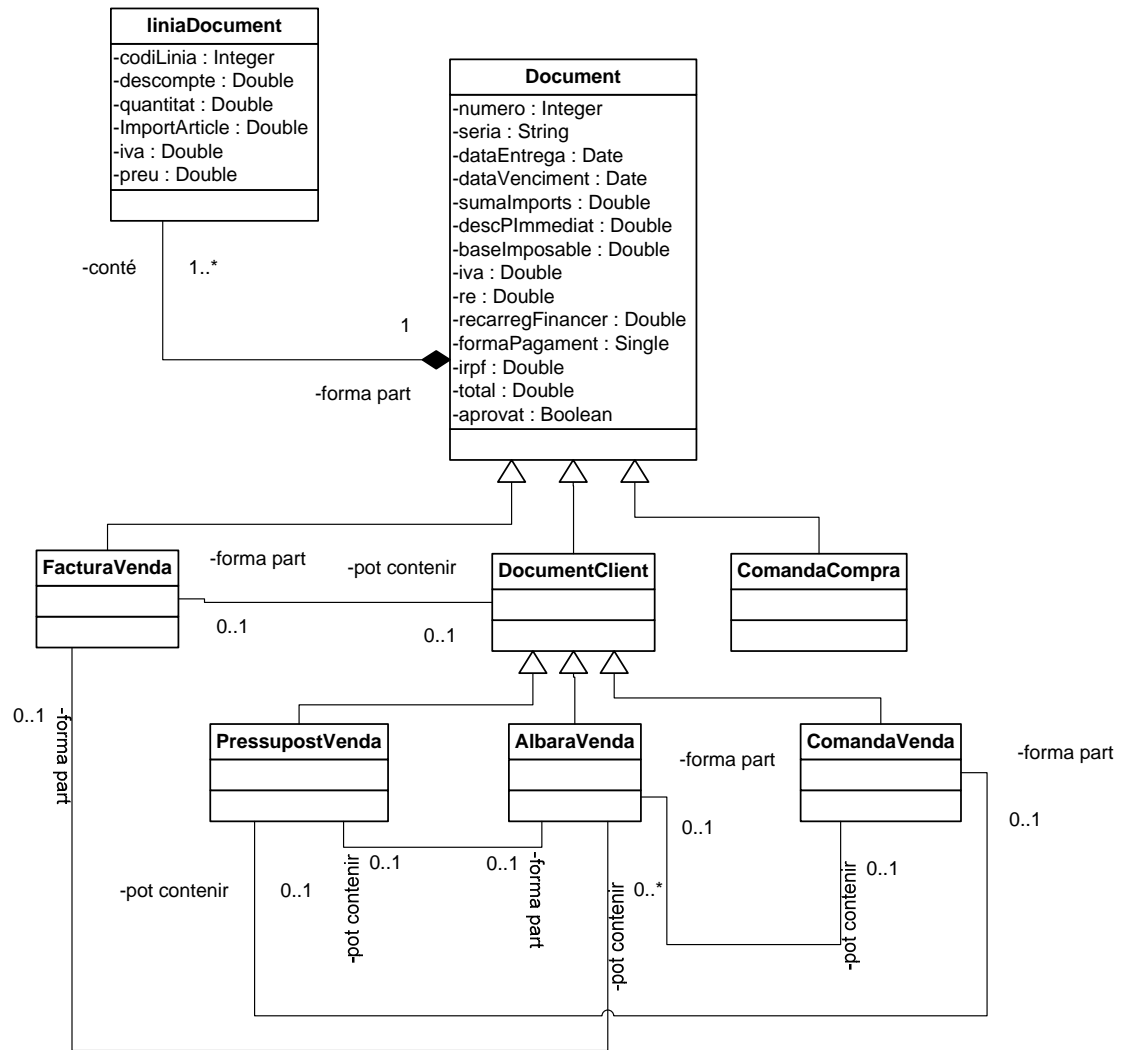
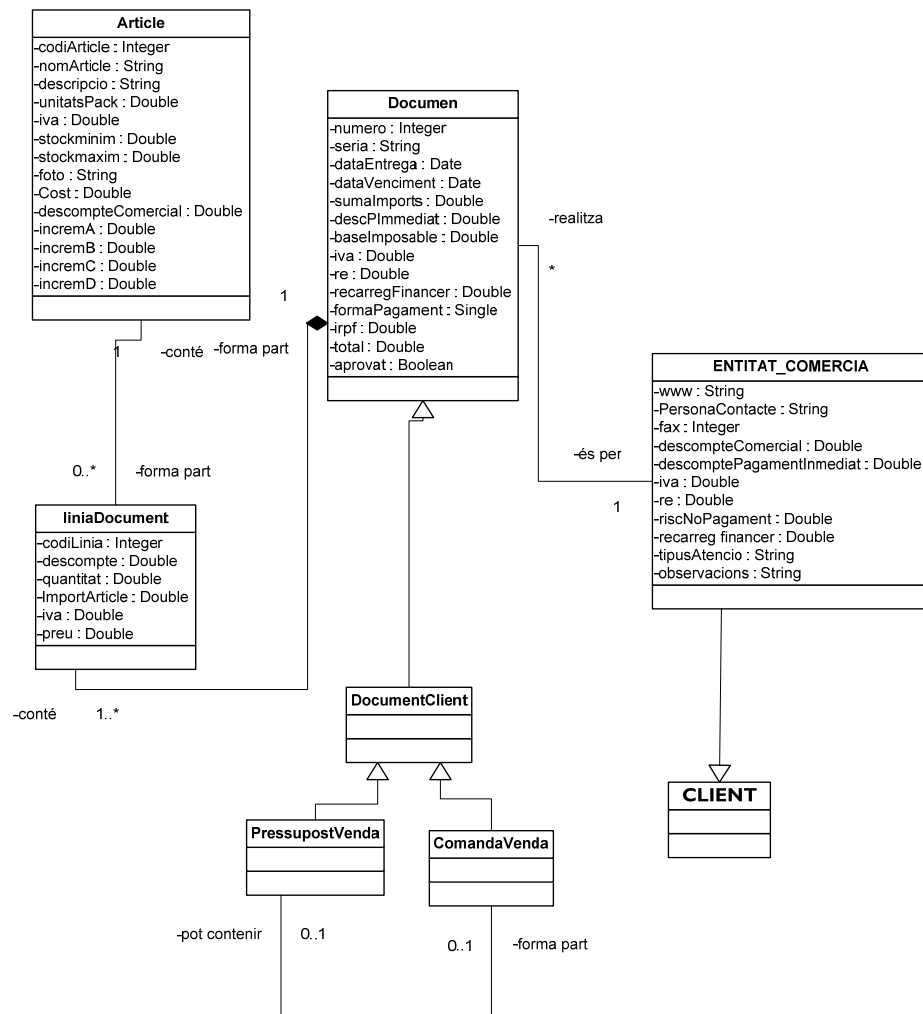


Diagrama de classes de l'herència de les classes que fan referència a documents



**Generació de comandes de clients.**

El diagrama que es mostra en la figura anterior mostra les classes implicades en la generació de una comanda on es mostra que pot estar relacionada amb un conjunt de línies per herència i aquestes cada una a un article, o pot estar relacionada a un pressupost. Per herència està relacionada amb una entitat\_comercial.

Cal tenir en compte que per crear una comanda sempre haurà d'estar relacionada a una entitat\_comercial, en aquest cas a un client. Si la comanda esta relacionada amb un pressupost aquest haurà d'existir dins del sistema. En el cas de fer-la directament, es a dir, relacionar-la amb línies, primerament crearem el document i tot seguit s'hauran de crear les línies, degut a la relació de composició.

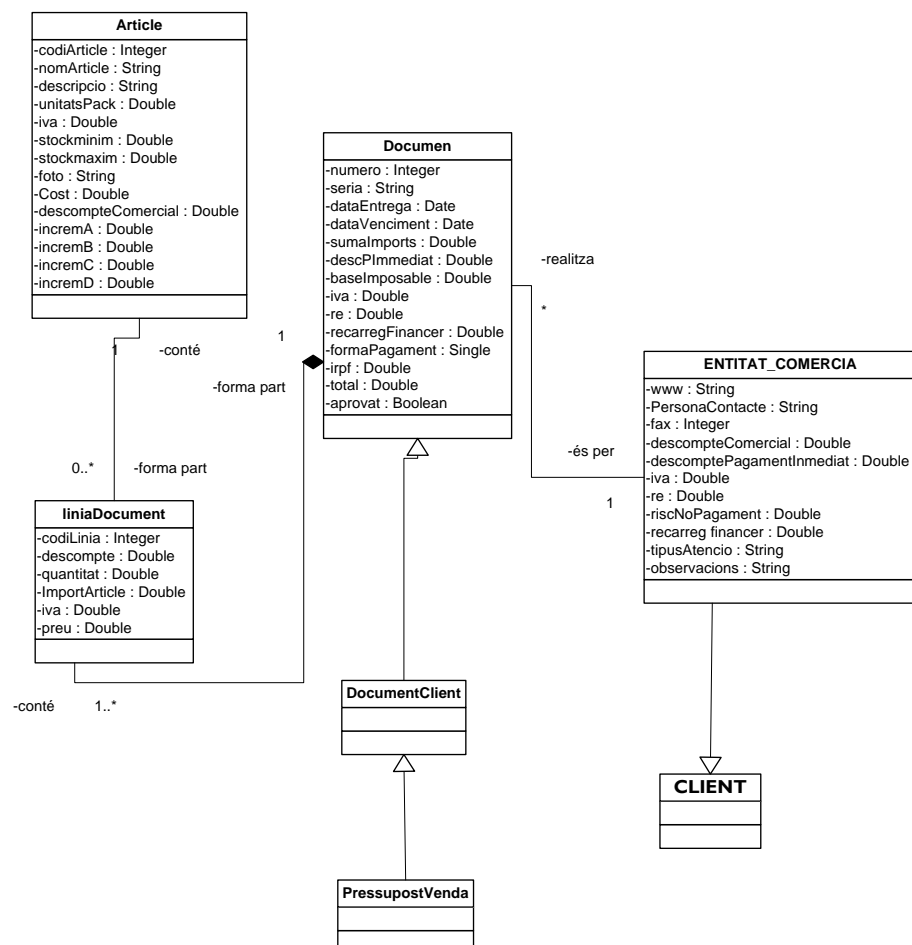
En el moment d'afegir articles a les línies, l'article haurà d'existir prèviament al sistema.

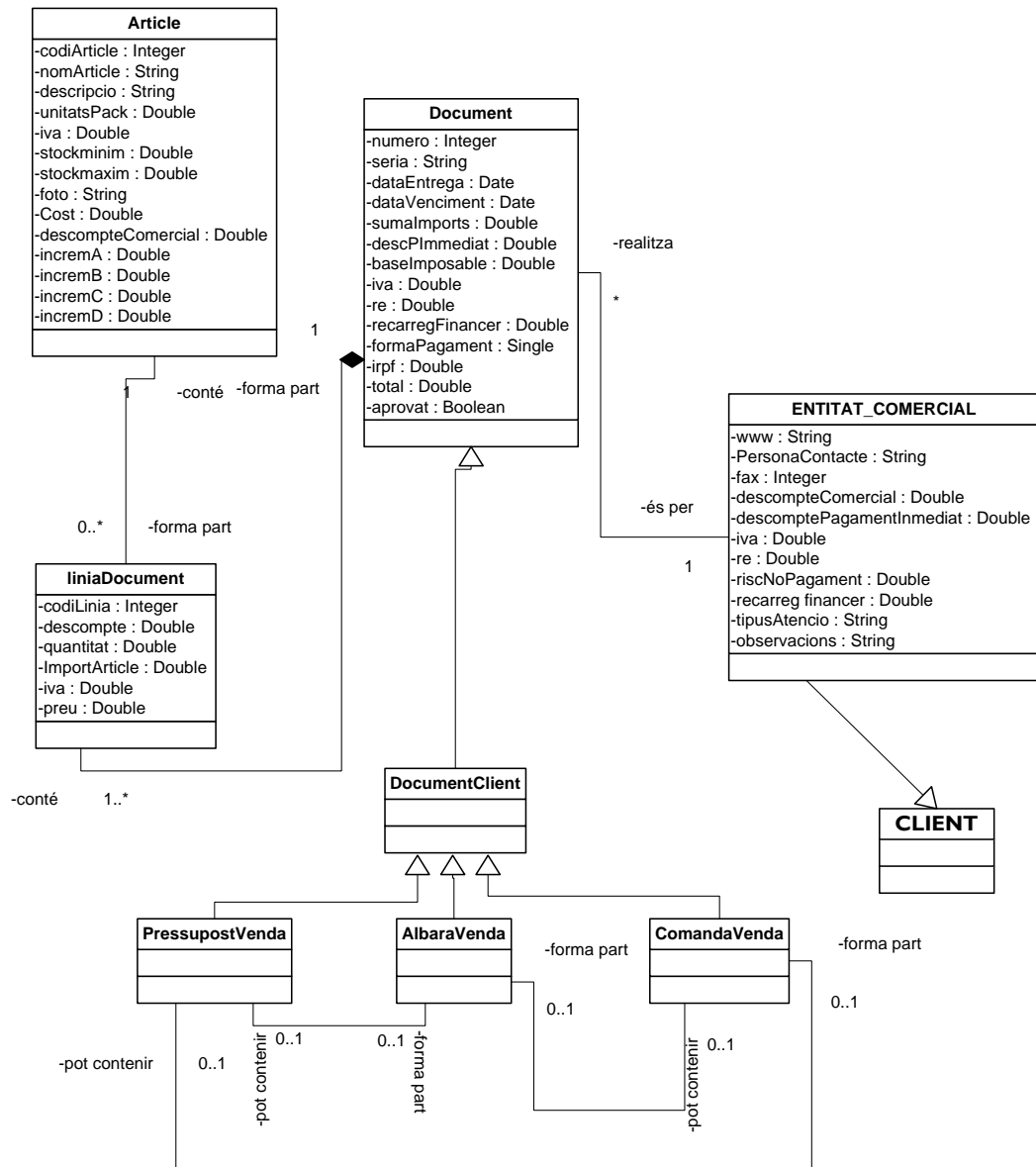
### **Realització de pressupostos.**

Per la realització de pressupostos les classes implicades són les que es mostren en la figura que hi ha tot seguit.

Veiem que per herència la classe PressupostVenda té associada d'una a un conjunt de línies i aquestes cada una a un article i a una entitat\_Comercial. Una entitat\_Comercial pot tenir associats molts documents de tipus pressupost.

Com en el cas de comanda un pressupost farà referència a una entitat\_comercial(client) ja existent al sistema, el document és crearà i tot seguit es crearan les línies de Document que se li associaran, per la seva relació de composició. En afegir un article a la líniaDocument aquest haurà d'existir dins el sistema.



**Gestió de lliurament de producte o servei.**

Per afegir un pressuposVenda com podem veure a les relacions se podrà fer a partir d'una manera directa (relacionant amb líniesDocument) o a partir d'una comandaVenda o un PressupostVenda com indiquen les relacions de la figura anterior.

Com en el cas de realització de pressupost i generació de comanda de client l'entitat\_comercial haurà d'existir prèviament al sistema per assignar-la, les línies s'assignaran després de la creació del document i finalment per cada

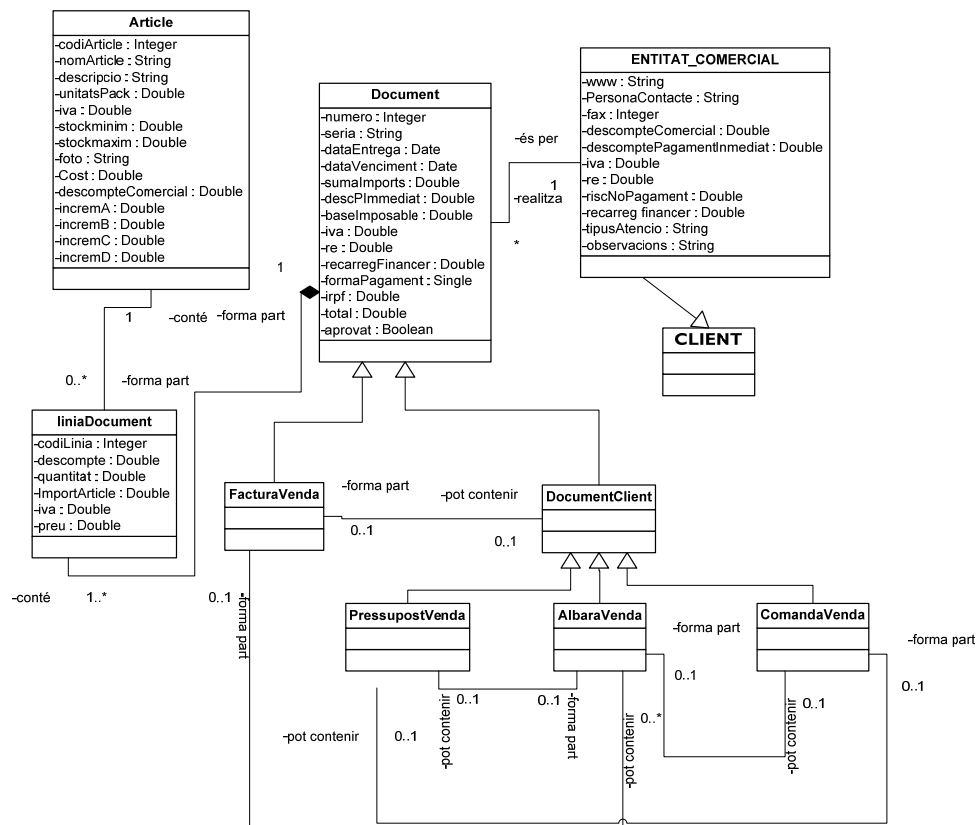
líniaDocument tindrà associat un article i cada article podrà estar en més d'una líniaDocument.

Un albarà podrà estar relacionat amb una comandaVenda o amb un PressupostVenda.

### Gestió de cobrament.

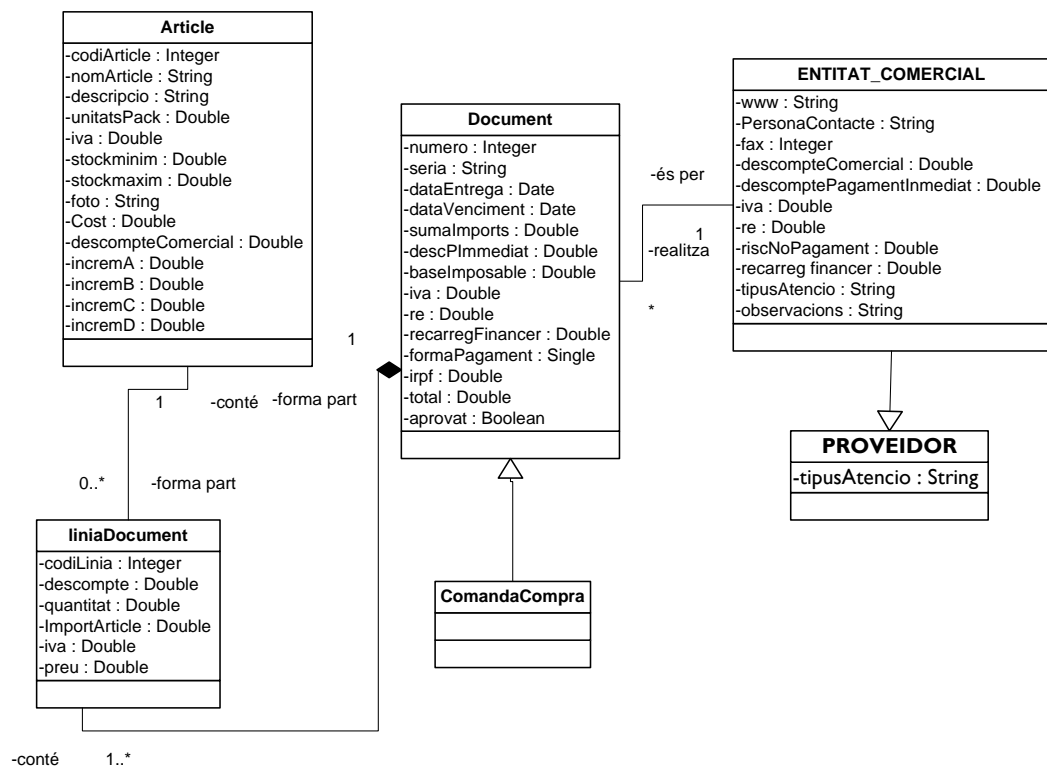
Com ja s'ha explicat en anterioritat (3 últims apartats) FacturaVenda té per herència la relació amb EntitatComercial i amb línia de factures.

En el moment de crear una FacturaVenda es podrà fer directament, es a dir, creant una o varies liniaDocument, a partir de fer la relació amb DocumentClient o a partir de la relació directe amb AlbaraVenda. La relació amb AlbaràVenda voldrà dir que una FacturaVenda podrà estar també relacionada amb molts Albarans.



**Gestió de devolucions.**

El model associat aquest cas es el mateix que l'anterior, ja que, la classe FacturaVenda també representa el document de devolució.

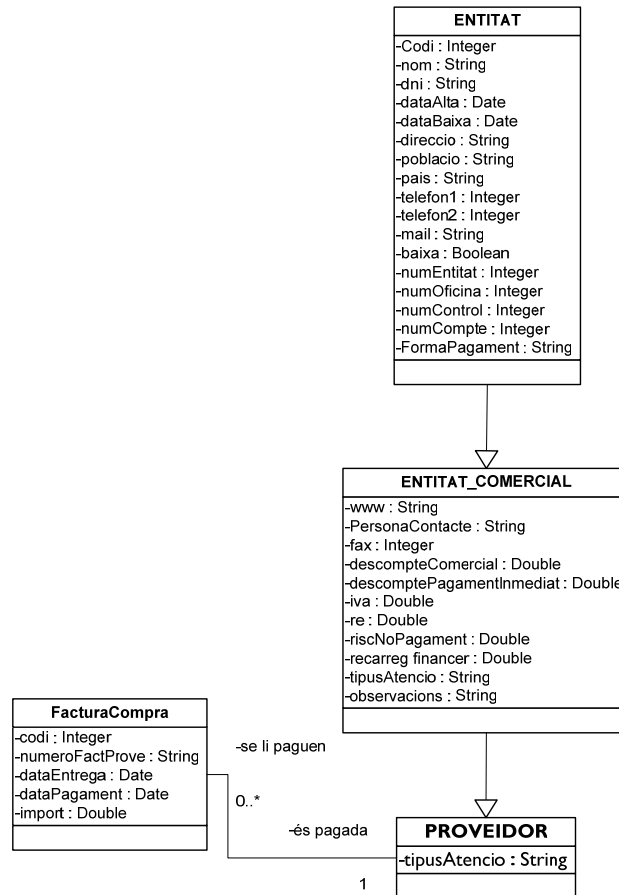
**Gestió de compres.****Gestió d'entrada de material.**

Per herència **ComandaCompra** esta relacionada amb **Entitat\_Comercial** i **liniaComanda**.

En la generació de la comanda sempre hi haurà d'haver relacionat una entitat\_comercial (tipus proveïdor) i una o un conjunt de **liniaDocument** on cada una d'elles tingui un article relacionat. La **liniaDocument** sempre estarà relacionada amb un **Document**.

(Pot ser que l'entrada de material no sigui necessària fer-la per ComandaCompar i directament es faci l'entrada modificant l'atribut stock.)

### **Gestió de pagaments.**



El model de classes associat aquest cas és l'anterior on la classe FacturaCompra està relacionada amb la classe Proveïdor de manera que una FacturaCompra sempre estarà relacionada amb un proveïdor existent en el sistema i a un proveïdor se li hauràn pagat cap o moltes factures.

### **Gestió de devolucions.**

El model associat aquest cas el mateix que l'anterior i també està associat al model que s'explicarà més endavant de control d'estoc de magatzem. Ja que una devolució pot implicar només un ingrés de diners on s'utilitzaria el cas anterior o només un control d'estocs o les dues casos.

### **3.5.4 Control d'estocs de magatzem.**

#### **Realització de comandes per a proveïdors.**

El model de classes d'aquest cas d'ús el mateix utilitzat en el casos d'ús Gestió d'entrada de material explicat anteriorment.

#### **Consultes d'estoc baix.**

La única classe implicada en la consulta d'estoc baix és Article, ja que, és la que guarda la quantitat d'articles que hi ha en magatzem i la quantitat d'estoc mínim i màxim que hi ha d'haver .

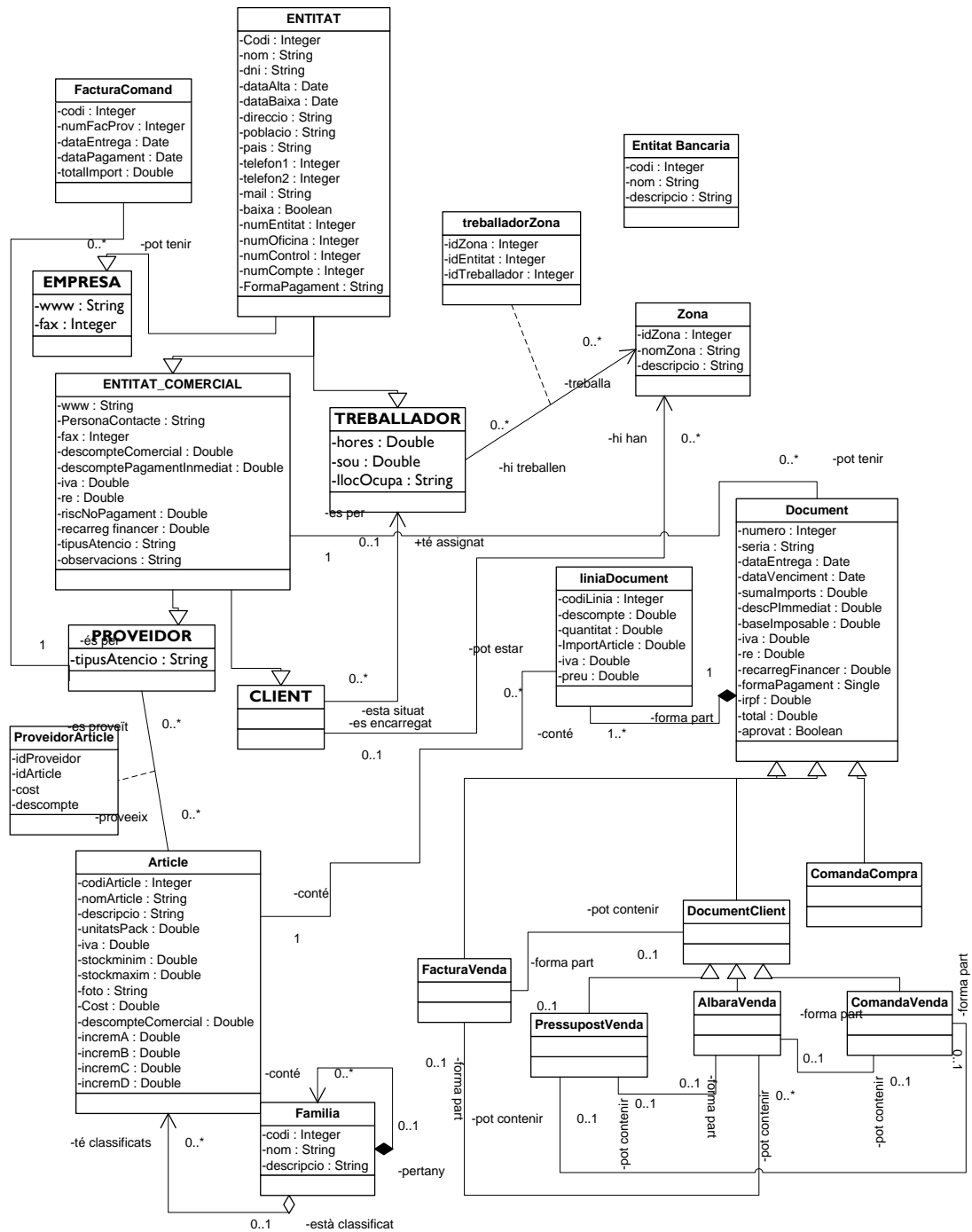
#### **Gestió d'estocs.**

Article
-codiArticle : Integer
-nomArticle : String
-descripcio : String
-unitatsPack : Double
-iva : Double
-stock : Double
-stockminim : Double
-stockmaxim : Double
-foto : String
-Cost : Double
-descompteComercial : Double
-incrementA : Double
-incrementB : Double
-incrementC : Double
-incrementD : Double

La única classe implicada en la consulta d'estoc baix és Article, ja que, és la que guarda la quantitat d'articles que hi ha en magatzem.

#### **Consultes**

Per a les consultes serà necessari tenir present tot el diagrama de classes que es presenta tot seguit





### **3.5 Definició d'interfície d'usuari.**

L'entorn del sistema a desenvolupar serà gràfic amb finestres, ja que la senzillesa d'utilització en aquest tipus d'interfície té un paper molt important i és un dels requeriments que es demanen.

#### ***Interfície gràfica***

La interfície gràfica estarà composta per una pantalla principal de l'aplicació que contindrà uns menús per accedir a les diferents funcionalitats de l'aplicació i una conjunt de botons per accedir a les funcions més importants i usuals de realitzar. Aquests botons de funcions usuals al accedir-hi mostraran un nou conjunt de botons que faran referència a les subfuncionalitats a poder realitzar.

Un cop seleccionada una de les opcions del menú s'obrirà una pantalla nova per poder realitzar la tasca seleccionada. En el cas de ser una gestió de dades físiques s'obrirà primer de tot una pantalla amb el llistat de totes les dades entrades i al costat un conjunt de botons amb les diferents opcions d'afegir, modificar, eliminar, consultar o activar i finalment el botó per sortir de la pantalla. En el cas que hi hagi alguna funció que no s'ha de poder permetre el botó no s'haurà de veure.

En el cas de voler fer una modificació, consulta, eliminar o activar alguna de les dades, s'haurà de seleccionar la fila on es troba la dada i tot seguit prémer el botó corresponent a la funció a realitzar, en cas que la dada estigui donada de baixa en el moment de seleccionar la fila on es troba el programa generarà un missatge d'error.

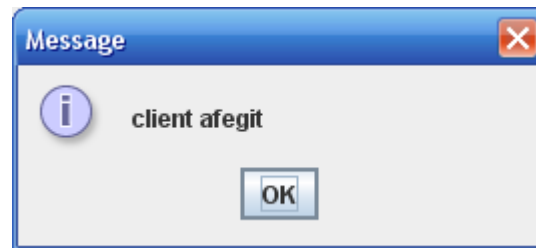
El botó eliminar una fila seleccionada a la taula, només ho permetrà si aquesta no té cap relació que pugui afectar les dades del sistema, en tot cas de no poder-ho fer sortirà una pantalla d'error informant de la possibilitat.

En els cas de afegir, modificar i consultar, s'obrirà una nova pantalla on es podran editar les dades escollides en el cas de la modificació o entrar noves dades en cas contrari, si per la qual cosa és una consulta els canvis que es facin no es guardaran. Un cop acceptem les modificacions o les dades entrades apareixerà

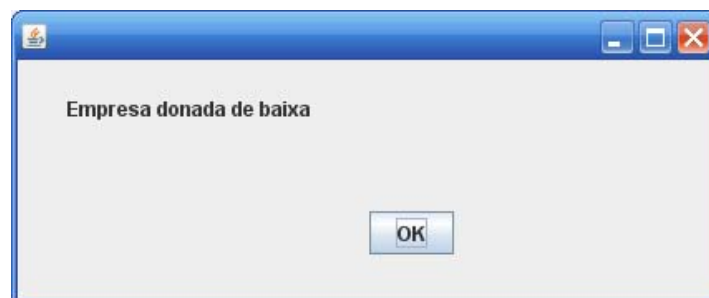
una pantalla d'informació confirmant l'acció feta i tot seguit només caldrà prémer el botó de sortir que ens tornarà a la pantalla de gestió d'aquell tipus de dades.

Els missatges d'error o de confirmació que poden sortir són els següents:

- Missatge de confirmació: el qual hi haurà un text dient que és el que es confirma.



- Missatge d'error o d'advertiment: el qual també anirà acompanyat d'un missatge amb l'error o l'advertència que s'està donant.



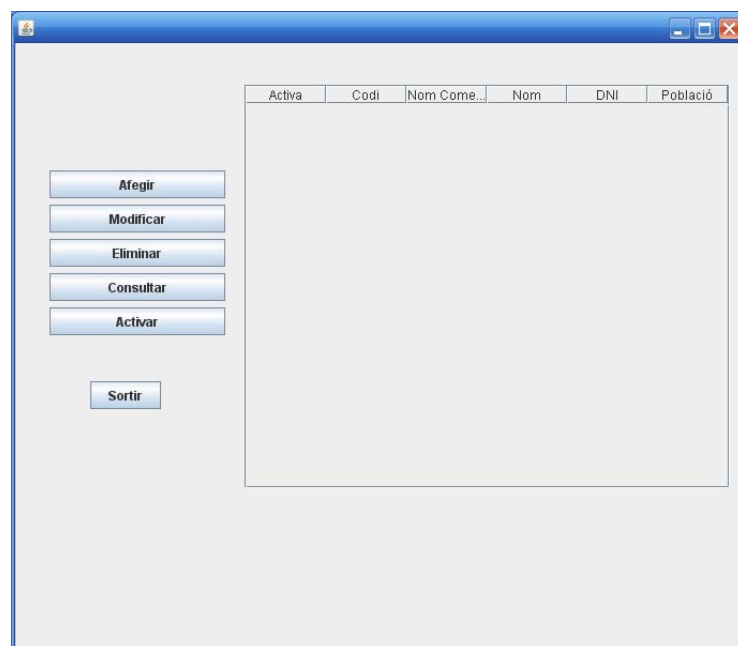
### 3.6.1 Especificació de forma individual de la interfície de pantalla.

Tot seguit s'especifiquen els formats d'interfície de pantalla per cada cas d'ús identificats en apartats anteriors (Especificació de casos d'ús)

#### 3.6.1.1 Gestió de dades físiques.

En cada una de la gestió de dades físiques, en excepció l'entrada o modificació de l'empresa, en accedir a la gestió primer de tot s'obrirà una pantalla, com la que es mostra tot seguit, amb el llistat de totes les dades entrades (segons la gestió escollida) i al costat esquerra els diferents botons de les diferents accions que es poden fer.

En el cas d'afegir només caldrà seleccionar el botó afegir i s'obrirà la pantalla de la dada a entrar i en cas de modificació o consulta caldrà seleccionar la fila on es troba la dada i seleccionar l'acció a realitzar i s'obrirà la pantalla amb la informació referent a la dada escollida. Si és vol eliminar o activar també s'haurà de fer el mateix que en la consulta o modificació, però no s'obrirà cap pantalla, únicament depèn del tipus de dada es modificarà el seu estat o en cas d'eliminar completament del sistema, la fila seleccionada s'eliminarà, donada aquesta opció s'intueix que la opció activar no serà visible.



**Gestió Empresa**

The screenshot shows a Windows-style application window titled 'Gestió Empresa'. The window contains a form with the following fields and sections:

- Left Column:**
  - Codi: [text box]
  - DNI: [text box]
  - Nom: [text box]
  - Data Alta: [text box]
  - Direcció: [text box]
  - Població: [text box]
  - País: [text box]
  - Telefons: [text box] [text box]
  - Fax: [text box]
  - www: [text box]
  - Mail: [text box]
- Bottom:**
  - Acceptar [button]
  - Sortir [button]
- Right Column:**
  - Nom Comercial: [text box]
  - Dades : [label]
  - Entitat Bancària [tab]
  - Codi de compte corrent: [text box] [text box] [text box] [text box]
  - Nom entitat bancària: [text box]
  - Nom de l'oficina: [text box]
  - Dades de l'oficina: [text box]
  - Persona de Contacte: [text box]

En aquest cas la primera vegada que és vulgui donar d'alta l'empresa la pantalla que s'obrirà serà l'anterior i en cas de que ja hagi estat donada d'alta s'obrirà la pantalla amb les dades referents a l'empres i les modificacions que es facin s'hauran de confirmar prement el botó acceptar i tot seguir prémer el botó sortir per tancar la pantalla. En cas que alguna dada no sigui correcte o necessària es generarà un missatge d'error dient que hi ha errors i marcarà amb vermell els quadres de text que contenen l'error, o són obligatoris.

**Gestió clients.**

The screenshot shows a web application interface for client management. It features a form with two columns. The left column contains fields for: Codi, NIF/CIF, Nom, Data Alta(dd/mm/aaaa), Direcció, Població, País, Telefons, Fax, www, and Mail. The right column contains fields for: Nom Comercial, Nom persona de contacte, and a section titled 'Dades:' with three tabs: CCC, D.economiques, and Assignacions. The CCC tab is active, showing fields for: Codi de compte corrent, Nom entitat bancària, Nom de l'oficina, Dades de l'oficina (a large text area), and Persona de Contacte. At the bottom, there are 'Acceptar' and 'Sortir' buttons.

Com es pot observar en la figura anterior el format pantalla es el mateix que en la gestió d'empresa però amb la diferencia que hi ha un panell amb 3 pestanyes que prement sobre d'elles podem veure la informació a entrar, modificar o consultar classificades segons tipus d'informació.

La primera pestanya es per entrar les dades bancàries que a mida que s'entra a cada una d'elles si l'entitat bancària està entrada al sistema els quadres de text que hi ha tot seguit (a sota) s'aniran emplenant amb la informació entrada de l'entitat.

En D. econòmiques és on es podran entrar totes les dades referents a descomptes, impostos i riscos econòmics. Les dades que caldrà escollir és la forma de pagament que utilitza el client i quina tarificació se li aplica. Cal recordar que si un producte no té diferents tarifes sempre s'aplicarà la Tarifa A.

Per acabar a la última pestanya trobem la opció de poder d'assignar el client a la zona on es troba, també si se li assigna podrà assignar algun treballador de tipus comercial i finalment si cal guardar alguna informació més o observació.

**Gestió proveïdor.**

És el mateix tipus de pantalla que en la gestió de clients. En aquest cas podem veure que té el mateix panell amb les tres pestanyes, on la primera com en el cas de client surten les dades bancàries amb el mateix funcionament que si l'entitat bancària entrada és existent va sortint la informació referent a ella.

En la segona pestanya (la que es pot veure en la figura) hi ha les dades econòmiques necessàries de tenir entrades del proveïdors amb una llista per triar quina es la forma de pagament que utilitzarà.

La pestanya assignacions hi trobarem una llista on assignarem quina es l'atenció amb la que s'ha de dirigir al proveïdor i una caixa de text on entrar altres observacions o informacions a destacar.

**Gestió Treballador.**

Com en els casos anteriors la pantalla que es mostra és la mateixa amb les caixes de text necessàries per entrar la informació necessària a saber de la persona i el panell amb les tres pestanyes on la primera pestanya com en els casos anteriors fa referència a la conta bancària, la segona a les dades econòmiques on en treballadors hi trobarem per entrar quines són les hores que treballa, el sou que cobra i la llista on es podrà triar la forma de pagament.

La tercera pestanya, referent a les assignacions, podrem assignar a quines zones treballa i quin es el lloc que ocupa a l'empresa (es pot veure en la figura següent).

The screenshot shows a software window with a tabbed interface. The 'Assignacions' tab is active. On the left, there is a form with the following fields: 'Codi' (with value '1'), 'NIF', 'Nom', 'Data Alta' (with value '1/9/2008'), 'Direcció', 'Població', 'País', 'Telefons', and 'Mail'. Below the form are two buttons: 'Acceptar' and 'Cancel·lar'. On the right, there is a panel titled 'Assignar Zona' with a large empty box for zone assignment. Below this, there is a label 'Lloc de ...' and a list box containing the following items: 'Administratiu', 'Magatzem', 'Comercial', and 'Repartidor'.

### **Gestió d'articles.**

Alhora d'entrar, modificar o consultar articles obtindrem una pantalla com la següent on es podran entrar la informació de l'article com el nom, descripció i una imatge. Com en el cas de les gestions de clients, proveïdors i treballadors tenim un panell amb tres pestanyes. La primera podrem donar entrar les dades com l'iva aplicat a l'article, els nivells que s'ha de trobar l'article en el magatzem i com es troba empaquetat.

La segona pestanya com la que es mostra a la figura següent s'hi podrà assignar els proveïdors que ens venen el producte, especificant el cost , el descompte i el preu net que acaba costant. En el moment que s'entren preus es va recalculant. També es podrà eliminar el proveïdor de l'article seleccionant prèviament la fila on es troba el proveïdor i prement el botó Eliminar. Finalment en aquesta pestanya podrem assignar un proveïdor com a habitual de la mateixa manera que s'elimina un proveïdor de la llista de proveïdors d'article. En el cas de voler eliminar un proveïdor que es assignat com a habitual el programa no ho deixarà fer i llençarà un missatge d'error informant de que no es pot realitzar l'operació.

Per acabar la pestanya Tarifa es podrà entrar els preus que pot tenir l'article en les quatre diferents tarifes que permet entrar el programa. En el cas d'haver assignat un proveïdor com habitual els quadres de text cost, descompte i cost net quedaran omplerts. A mida d'entrar o canviar imports dels diferents quantres de text s'anirà calculant els altres, per entendre-ho millor si entrem un cost i un descompte és calcula el cost net o si s'entra el cos i tot seguit el net es calcula el descompte aplicat. En tenir el quadre cost net omplert la primera tarifa sortirà el cost només s'haurà d'indicar el marge que s'afegeix i es calcularà el preu final i el preu més iva, o directament entrar el preu final i es calcularà el marge i l'iva.

### **Gestió zones.**

En aquest cas en el moment d'afegir de nou una zona només es podrà entrar el nom i la descripció, però si es vol modificar la zona llavors és podrà fer l'assignació corresponent de treballadors i clients a la zona.

En aquesta pantalla tenim un panell amb dues pestanyes, la primera per fer l'assignació de treballadors a les diferents zones, on es pot fer un filtre dels treballadors que no estan assignats a cap zona. Es podrà fer les pertinents



assignacions i desassignacions seleccionant prèviament la fila on es troba el treballador i prémer el boto que indica l'acció a voler realitzar.

A la pestanya clients podrem fer l'assignació de clients a la zona que s'està modificant. Només es visualitzaran els clients que no tenen cap zona assignada.

The screenshot shows a software window with a blue title bar and standard Windows window controls. It has two tabs: 'Treballadors' (selected) and 'Clients'. On the left, there is a form for a worker with fields for 'Codi' (with a dropdown arrow), 'Nom' (containing 'asfdsadf'), and 'Descripció' (containing 'asfdsadf'). Below these are 'Acceptar' and 'Sortir' buttons. The main area is divided into two panes. The left pane, titled 'Treballadors assignats a la zona', is empty. The right pane, titled 'Treballadors NO assignats a la zona', contains a radio button labeled 'Marcar per només veure treballadors no assignats a cap zona'. Between the two panes are 'Desassignar>' and '<Assignar' buttons.

### ***Gestió Entitat Bancària.***

En el moment de afegir, modificar o consultar una entitat bancària se'ns mostrarà la pantalla següent on es podrà entrar la informació necessària.

The screenshot shows a software window with a blue title bar and standard Windows window controls. It contains a form for a bank entity with the following fields: 'Codi Entitat Bancària:', 'Nom Entitat bancària:', 'Codi Oficina:', 'Nom de l'oficina:', 'Direcció:', 'Població:', 'País:', 'Persona Contacte:', 'Telèfon 1:', 'Telèfon 2:', and 'Fax:'. Each field has a corresponding text input box. At the bottom right are 'Acceptar' and 'Sortir' buttons.

**Gestió de famílies.**

The screenshot shows a window titled "GESTOR D'ALTES". On the left, there are three input fields: "Codí:" with the value "1", "Nom:", and "Descripció:". Below these fields are two buttons: "Sortir" and "Acceptar". On the right, there is a panel titled "Tria en quina Família es subclassifica" containing a large empty rectangular box for selection.

En la gestió de famílies es podrà entrar la informació necessària a guardar de les famílies i en cas de ser una subfamília s'haurà de seleccionar de la llista que es mostra a la dreta de la pantalla la família a la que formarà part la subfamília.

**Gestió de vendes**

Com en el cas de les dades físiques, en el moment de fer la gestió de qualsevol tipus de document, sigui pressupost, comanda, factura o albarà, quan accedim a la opció es mostrarà una pantalla amb el llistat tots els documents generats i al costat les diferents opcions que es poden fer.

Cal dir que en cas de fer modificacions, si el document no està relacionat amb cap altre document ens mostrarà la pantalla de generar directament un document de tal manera que es podran fer les modificacions requerides. En cas d'estar associat a un document es mostrarà el document sense la possibilitat de ser modificable.

En escollir consulta es mostrarà la pantalla de generar documents amb totes les dades que formen el document.

Per eliminar un document es podrà fet només si aquest no esta aprovat o realitzat i en el cas de les factures no hi haurà l'opció d'eliminar, però si la de cobrar, com en el cas de les devolucions.

Pressupost

Número pressupost 1

Codi client

Data Entrega:

Data Venciment:

Descompte pagament immediat: ☐ %

Afegir article:

Codi	Descripció	Quantitat	IVA	Preu	Descompte	Preu Net	Import
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Eliminar l'article, previament havent seleccionat la fila:

Base Imposable  IVA  RE  Total

Fer generar qualsevol document ens sortirà aquesta pantalla on el primer item a entrar és el codi de client, que en cas de no saber-lo mitjançant el botó buscar obtindrem una llista de tots el clients actius i es podrà seleccionar. Tot seguit al costat dret tenim les dates d'entrega i de venciment del document que per defecte surt la del dia que es troba en el moment d'entrar el document i la data de venciment un mes següent.

Tot seguit es poden entrar els articles entrant el codi o mitjançant el botó buscar s'obre una pantalla on s'escull el producte, un cop escollit el producte o entrat el codi automàticament s'omplen els quadres de text següents segons l'article a entrar i el client a qui va dirigida la comanda, per defecte la quantitat de producte és 1, les quantitats es podran canviar. Per afegir tant sols caldrà prémer el botó amb el signe + i per eliminar-lo del document seleccionar la fila eliminar i prémer el botó amb el signe negatiu. Podrem apreciar que cada vegada els totals del document es van canviant.

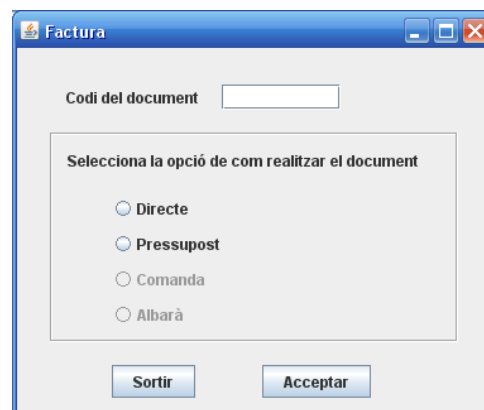
Si l'article que s'afegeix a la factura fa que l'estoc passi del mínim sortirà una pantalla d'advertència, de la mateixa manera que si la factura és negativa (devolució) si sobrepassa el nivell màxim d'estoc també avisarà. Si la quantitat a

afegir fa que no quedi estoc l'article no s'afegirà al document i llençarà un missatge d'advertència que no hi ha prou estoc.

Per finalitzar el document i donar-lo per vàlid només caldrà prémer l'acceptar, tot seguit sorgirà una pantalla amb la confirmació de l'alta del document, si no hi ha cap producte entrar o cap client no deixarà donar d'alta el document.

### ***Generació de comandes de clients***

Si es vol afegir una comanda primer de tot s'obrirà aquesta pantalleta que ens demana si es vol fer la comanda directament o a partir d'un pressupost. Si s'escull directament s'obrirà la pantalla explicada al principi d'aquest apartat. En el cas d'escollir a partir d'un pressupost, caldrà entrar el número de pressupost sense la seria i prémer l'acceptar, si el pressupost no esta aprovat s'obrirà la pantalla amb el document i per confirmar la generació de la comanda només caldrà acceptar el document

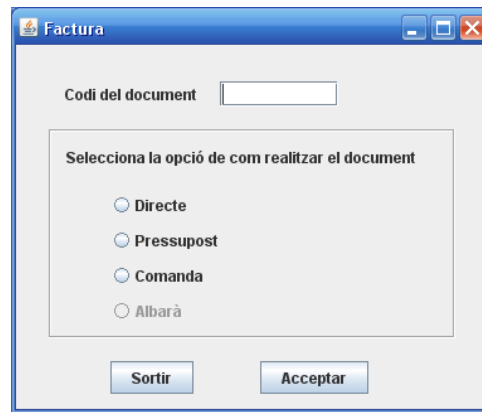


### ***Realitzar pressupost***

Un pressupost només es podrà realitzar directament de manera que per afegir el pressupost, la pantalla que es mostrarà serà la realitzar un document nou. (Pantalla mostrada al principi d'aquest apartat).

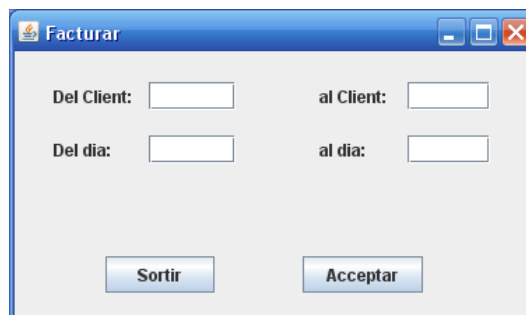
### ***Gestió de lliurament de producte o servei.***

En aquest apartat és on es passarà a realitzar un albarà de manera que primer s'haurà d'escollir a partir de quin punt es vol fer.



Per tant es podrà fer directament, on es mostrarà la pantalla de realització d'un document, ja explicat, a partir d'un pressupost o a partir de comandes.

Si es fa a partir d'un pressupost la metodologia a seguir e la mateixa que en la generació de comanda a partir d'un pressupost, en canvi si es fa a partir de comandes, és podrà fer, a partir de totes les comandes que no s'hagin servit i seleccionant un rang de client i un rang de dates.



Només caldrà seleccionar de quin client a quin altre és vol genera les comandes i entre quines dates (si només es vol fer d'un client, entrant el mateix codi de client als dos quadres de text n'hi haurà prou). Si falta alguna dada a entrar el programa llençarà una advertència de falta de dades.

Un cop feta la selecció es mostraran en una nova pantalla com la següent totes les comandes que s'han trobat dins els parametres entrats.

Factures Generades

Del client:  al client:  Data d'entrega:

De la data:  a la data:

Codi document	Client	limport
---------------	--------	---------

Sortir Acceptar

Per generar els albarans només caldrà entrar la data d'entrega de l'albarà i prémer el botó acceptat. Tot seguit el programa generarà un albarà per cada una de les comandes entrades. Un cop finalitzat el procés, la taula s'actualitzarà amb els documents generats i desapareixerà el botó acceptar.

### ***Gestió de facturació***

Per fer la facturació es passos a seguir seran els mateixos que la realització d'albarans amb les mateixes opcions, la diferència està en que les factures és poden fer a partir d'albarans.

Els passos a seguir per realitzar factures a partir d'albarans són les mateixes que realitzar albarans a partir comandes, internament funciona diferent, ja que per cada client genera una factura amb els albarans pendents de facturar. Les pantalles que es veuran seran les mateixes també.

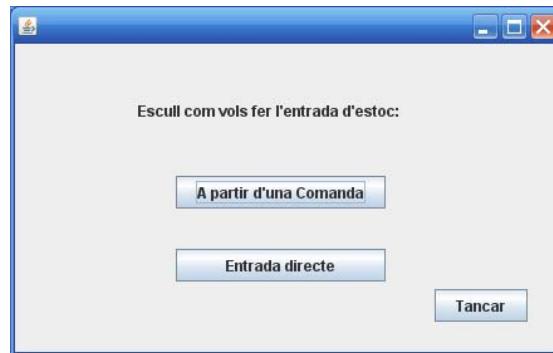
### ***Generació de devolució***

Com que una devolució no es pot generar a partir de cap document es farà directament, cal tenir en compte que els preus s'hauran de canviar a negatius. Una devolució no deixa de ser una factura en negatiu.

## **Gestió de compres**

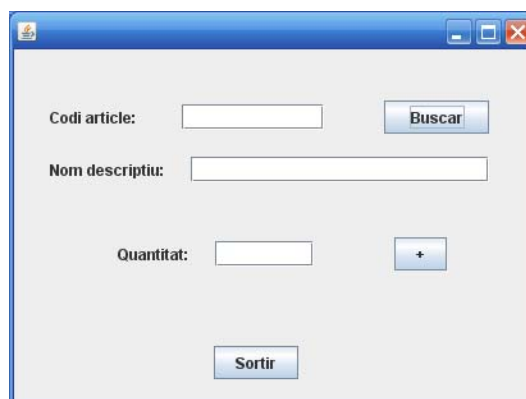
### **Gestió d'entrada de material**

En seleccionar aquesta funció s'obrirà una pantalla que preguntarà com es vol fer l'entrada dels articles.



Si s'escull la primera opció s'obrirà la pantalla de gestió de comandes de compra on només caldrà seleccionar la comanda que és vol donar per entregada, automaticament el programa farà el control d'estoc de cada un dels articles que compon la comanda.

En el cas d'entrar els articles manualment la pantalla que és mostrarà serà la següent:



On només caldrà entrar el codi de l'article o buscant-lo a partir del botó buscar i escriguent la quantitat d'article, prement el botó de signe positiu es farà l'actualització.

### **Gestió de pagaments**

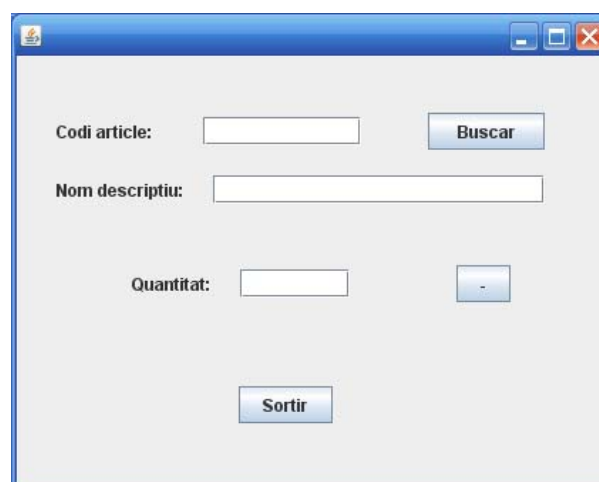


The screenshot shows a software window titled "Gestió de pagaments". It features several input fields and buttons. At the top, there is a "Codi proveïdor:" label followed by a text box and a "Buscar" button. Below this is a "Nom:" label followed by a text box. Further down, there are three stacked input fields labeled "Número de la factura:", "Data Entrega de la factura:", and "Data Pagament de la factura:". At the bottom of these fields is an "Import total:" label followed by a text box. At the very bottom of the window, there are two buttons: "Acceptar" on the left and "Sortir" on the right.

Per entrar les factures que ens envia un proveïdor, s'haurà d'entrar el codi de proveïdor a buscar-lo mitjançant el botó buscar, entrar el número de factura, la data d'entrega de la factura i l'import total, en cas de no entrar la data de pagament voldrà dir que aquesta factura no ha estat pagada al proveïdor. En prémer el botó acceptar quedarà entrada la factura, si aquesta ja ha estat entrada el programa llençarà un missatge dient que aquesta factura ja ha estat entrada.

### **Gestió de devolucions**

La primera pantalla que ens mostra és la mateixa que en la gestió de pagaments, on s'haurà d'entrar les mateixes dades (l'import total també s'haurà d'entrar en negatiu), un cop acceptada l'entrada, si aquesta devolució no existeix en el sistema sobrirà una pantalla per fer sortida d'estoc com la següent.



The screenshot shows a software window titled "Gestió de devolucions". It features three input fields and two buttons. At the top, there is a "Codi article:" label followed by a text box and a "Buscar" button. Below this is a "Nom descriptiu:" label followed by a text box. Further down, there is a "Quantitat:" label followed by a text box and a small button with a minus sign. At the bottom of the window, there is a "Sortir" button.



Aquesta pantalla té el mateix funcionament que la gestió d'entrada d'articles directament.

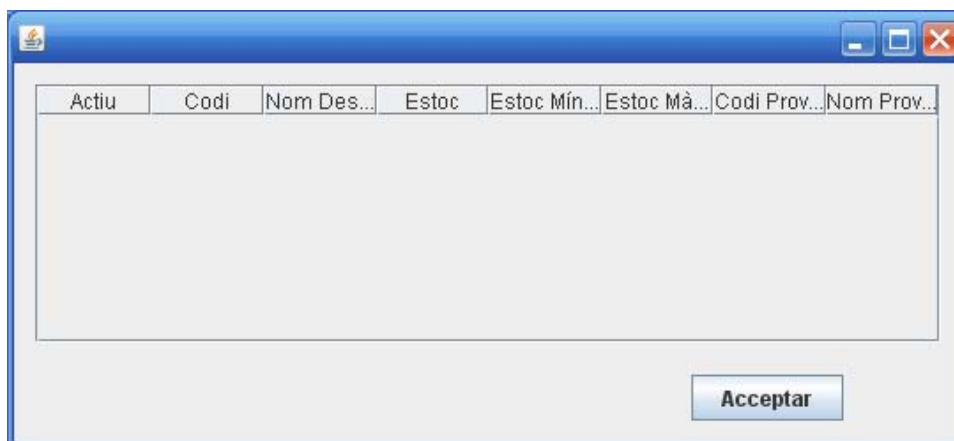
### **Control d'estocs de magatzem**

#### **Realització de comandes per a proveïdors**

Les pantalles i les passes a seguir són exactament les mateixes que en la generació d'un pressupost, la diferència està que el control d'estoc no es fa fins que no es valida la comanda. Es a dir que la pantalla inicial que es mostrarà es la de gestió de documents on sortirà el llistat de totes les comandes generades i on es podrà fer les gestions d'afegir (igual que un pressupost, directament), consultar, modificar i eliminar sempre i quan no hagi estat validada i validar la comanda.

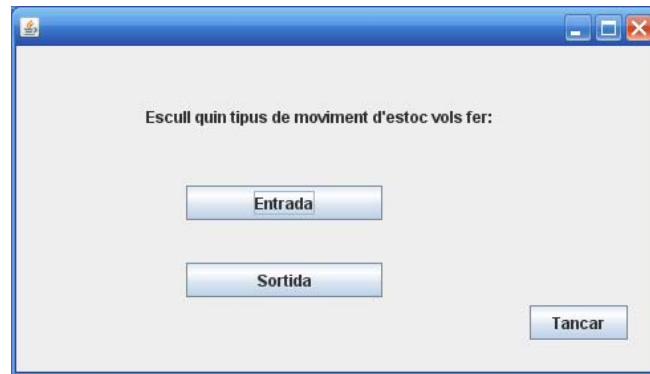
#### **Consultes d'estoc baix**

En accedir en aquesta opció sortirà una pagina amb una taula on es podrà veure la llista d'articles que es troben per sota l'estoc mínim.



#### **Moviment d'articles**

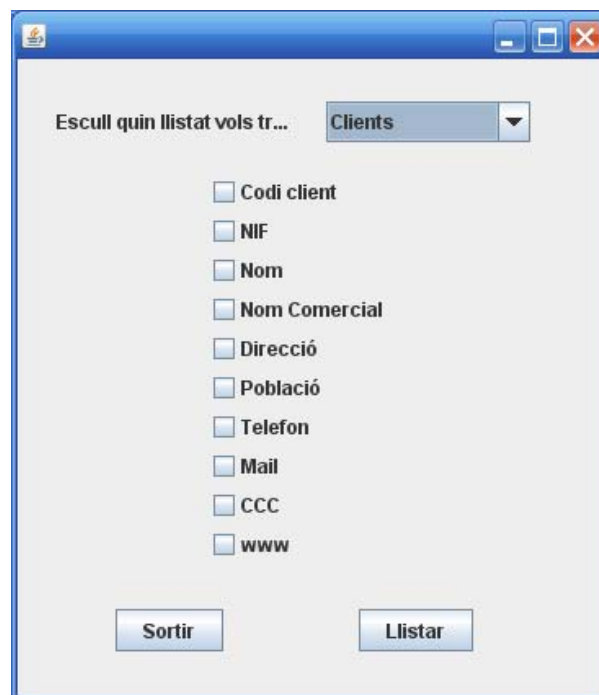
Els articles es podran entrar o sortir de manera que la primera pantalla que es mostra en escollir aquesta opció és la següent on s'haurà d'escollir si es vol fer una entrada o una sortida, si s'escull la d'entrada, s'obrirà la pantalla ja explicada en la gestió d'entrada de material i si s'escull fer una sortida s'obrirà la pantalla explicada en gestió de devolucions de compra.



## ***Resum d'ingressos i Despeses i Consultes***

### ***Llistat***

Per poder llistar les diferents dades físiques que tenim entrades podem accedir fer-ho a partir d la pantalla següent:



A partir d'escollir sobre quines dades es vol fer el llistat s'actualitzarà la informació de la dada, només caldrà escollir quins camps són els que es volen veure a la llista. Prement el botó acceptar s'obrirà una pantalla amb el llistat seleccionat i la informació que prèviament s'ha demanat que es llisti.

### **Consulta sobre Clients**

En la consulta sobre clients, escollint amb el botó buscar sobre quin client es vol fer la consulta, sortirà el resultat del llistat de factures que se li han generat, el llistat dels productes que ha comprat, el total facturat i el total de deutes que té, també es mostra el gràfic amb els ingressos que s'ha fet cada any gràcies aquest client.

The screenshot shows a Java Swing window titled 'Consulta sobre Clients'. It contains the following elements:

- A label 'Codi del Client' above two empty text input fields.
- A 'Buscar' button in the top right corner.
- A label 'Llistat de factures:' above a large empty rectangular area for displaying a list.
- A label 'Llistat de productes que compra:' above another large empty rectangular area for displaying a list.
- Two labels with corresponding input fields: 'Total facturat:' and 'Total deutes:'.
- A 'Sortir' button in the bottom left corner.
- A label 'jLabel20' centered near the bottom.

### **Consulta sobre Proveïdors**

En accedir aquesta consulta com en el cas de consulta sobre client també s'obrirà una pantalla igual on triant amb el botó buscar el proveïdor s'obtindrà el llistat de factures pagades, el llistat de factures en deute, el total facturat i el total que se li té a deure. I es visualitzarà un gràfic amb el total de despeses generades cada any amb aquest proveïdor.

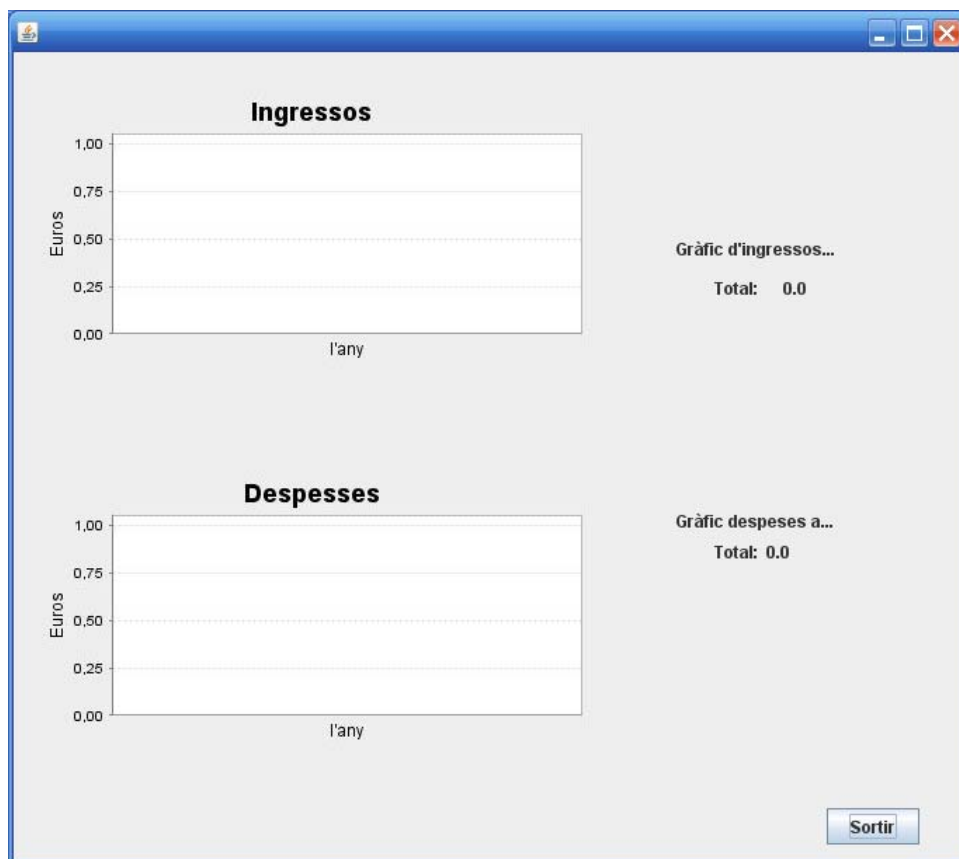
### **Consulta sobre articles**

La pantalla que s'accedirà serà igual que l'anterior però la diferència és que la primera llista serà dels 10 articles més venuts i la segona llista es veurà per cada

article la quantitat d'articles que s'han comprat als proveïdors, que s'ha entrat per comanda.

### ***Consulta d'Ingressos i despeses***

En accedir aquesta opció es veurà un gràfic amb el total d'ingressos generats i un altre amb el total de despeses generades durant cada any, al costat de cada gràfic el total.



## **4.Disseny del sistema d'informació**

En aquest capítol es veuran les activitats següents de disseny:

- Definició de l'arquitectura del sistema.
- Disseny de l'arquitectura de suport.
- Disseny de casos d'ús reals.
- Disseny de classes.
- Disseny físic de dades.
- Generació d'especificacions de construcció.

### **4.1 Definició de l'arquitectura del sistema.**

#### **4.1.1 Definició de nivells d'arquitectura**

Els nivells de l'arquitectura del sistema a desenvolupar és senzilla, ja que es tracta d'un entorn monousuari i no serà necessari la utilització d'arquitectura com podria ser la de client-servidor, que s'hauria d'utilitzar en el cas que l'aplicació pogués ser multiusuari, que passaria a necessitar un servidor per el control d'accés a la base de dades des de diferents estacions de treball.

Com que l'aplicació i la base de dades es trobaran en el mateix equip els nivells d'arquitectura serà d'un sol nivell. Per tant en el Client serà on es trobaran tots els components

- La base de dades → base de dades del sistema.
- El gestor de la base de dades → gestor de comunicació entre la base de dades i els objectes de negoci.
- D'interfície d'usuari → sistema de comunicació amb l'usuari.
- Objectes de negoci → objectes de negoci que encapsulen la informació de la base de dades.

Es en aquest punt on cal esmentar que l'arquitectura que s'utilitzarà per la creació de projecte, en aquests cas es una arquitectura de programació per capes on l'estructuració serà la següent:

- Capa de la base de dades → capa encarregada de portar a terme la negociació entre la base de dades i la capa d'entitats. Utilitza les eines del gestor de la base de dades per la comunicació.
- Capa d'entitats → capa que contindrà els objectes que representaran les diferents entitats del sistema (classes que s'han vist en el capítol 3).
- Capa de control → capa que s'encarrega del negoci entre la interfície gràfica i les entitats.
- Capa d'interfície gràfica. → capa de construcció de d'interfície gràfica.
- 

#### **4.1.2 Identificació de requisits de disseny i construcció**

Per dur a terme el disseny dels sistema caldrà tenir en compte que l'arquitectura que s'utilitzarà serà una arquitectura de programació per capes, per tan caldrà tenir, en tot moment, molt clar quins són els límits de cada una d'elles.

Abans de començar el procés de construcció caldrà tenir en compte quin sobre quin sistema operatiu es desenvoluparà l'aplicació, quines són les característiques de l'equí de desenvolupament per tal que no hi hagi limitacions en el procés de construcció, el llenguatge emprat per el desenvolupament de l'aplicació i les eines de desenvolupament que s'utilitzaran.

#### **4.1.3 Especificació d'estàndards i normes de disseny i construcció**

La notació utilitzada en el disseny del projecte és la de UML.

En la construcció del sistema s'haurà de tenir en compte les següents normes d'anotació.

- Els components referents a la construcció d'interfície gràfica aniran acompanyats del nom de la pantalla i tot seguit del nom Pantalla.
- Els components referents a la capa de negoci entre d'interfície i les entitats aniran precedits per la paraula Control

- Els objectes que representen les entitats, per tant, els que es troben a la capa d'Entitats no tindran cap terminació
- Les classes que formen part de la capa de negoci amb la base de dades començaran tots amb la paraula BDD.
- El mòdul principal del programa s'anomenarà main.
- Els objectes es classificaran en diferents paquets, aquets paquets són la representació de les diferents capes.
- Els noms de les classes tots començaran en majúscules.
- Els noms de les funcions i mètodes començaran amb minúscula i si està compost per dues la següent serà amb majúscula.

#### ***4.1.4 Identificació de sistemes de disseny.***

Els sistemes de disseny coincideixen amb les capes presentades en l'apartat de nivells d'arquitectura.

- Capa de la base de dades
- Capa d'entitats
- Capa de control
- Capa d'interfície gràfica

#### **4.1.5 Especificació de l'entorn tecnològic**

Les especificacions de l'entorn tecnològic que s'hauran de tenir en compte en el desenvolupament del projecte són.

- Sistema operatiu: Microsoft Windows XP.
- Gestor de la base de dades: MySQL.
- Equip amb prestacions normals.

#### **4.1.6 Especificació de requisits d'operació i seguretat**

En tot moment en la construcció del projecte s'haurà de tenir molt present la protecció de d'informació del sistema, per tal que no hi hagi pèrdues d'informació. Per tant s'haurà de tenir present els següents requisits.

- En la modificació de les dades s'hauran de tenir en compte les regles d'integritat referencial. Per tal que no hi hagi incoherències en la informació emmagatzemada en la base de dades.
- Abans de processar la informació cap a la base de dades s'haurà de fer les pertinents comprovacions per tal que no s'emmagatzemin dades errònies.

### **4.2 Disseny de l'arquitectura de suport**

En el disseny de l'arquitectura de suport s'hauran de tenir en compte tot el que s'ha explicat en l'apartat 4.1 ressaltat sobretot que l'arquitectura utilitzada per la creació del projecte és una arquitectura per capes, les qual ja estan explicades quines són les seves funcions (4.1.1)

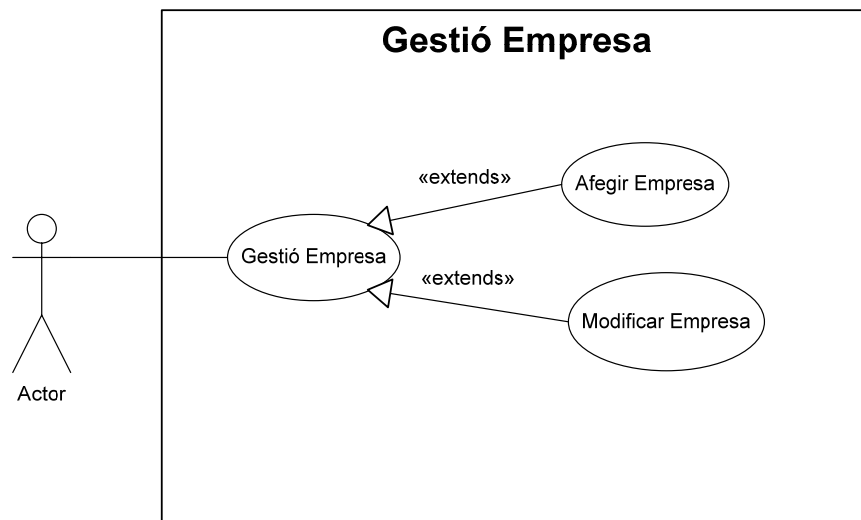


### 4.3 Disseny de casos d'ús reals

A partir dels casos d'ús que s'han identificat en el capítol tres, ara es passarà a fer el disseny dels casos d'ús reals identificant les classes associades a cada un d'ells.

#### ***Gestió de les dades físiques***

##### ***Gestió Empresa***



En la figura anterior es mostra el cas d'ús de gestió empresa. Es pot observar que d'aquest cas s'extenen dos més de simples que realitzen les accions pròpies d'una gestió de dades.

Afegir una empresa només es podrà fer una vegada, a partir del moment que per primera vegada s'hagi afegit una empresa les següents vegades només es podrà fer modificacions sobre l'empresa, ja que només a de permetre tenir una empresa entrada.

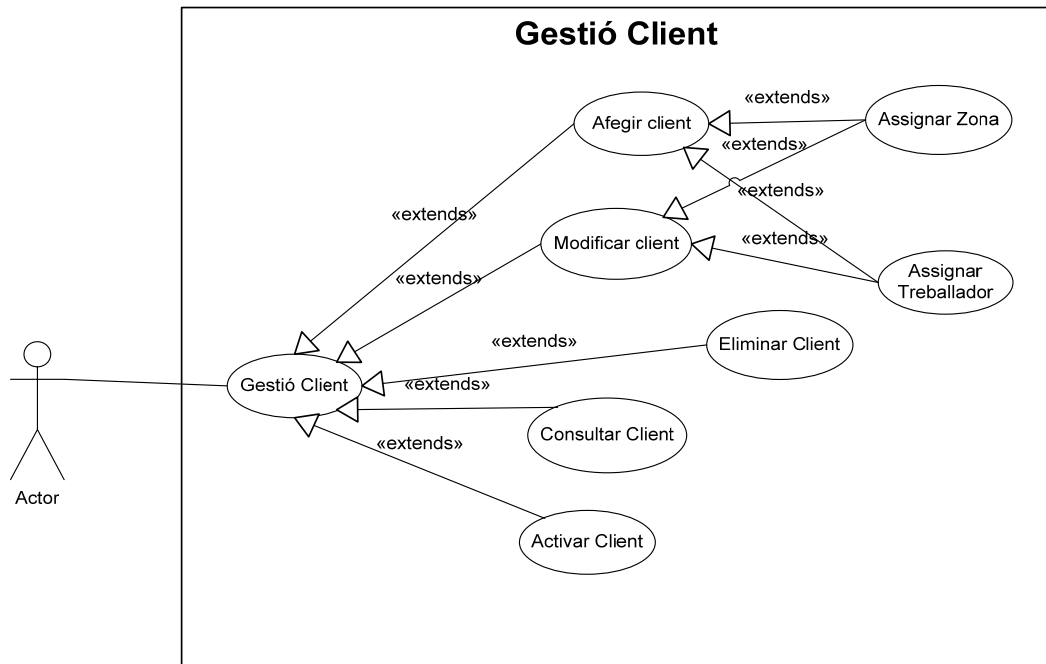
Per tant el primer moment d'afegir una empresa sempre s'haurà de comprovar que no n'hi hagi cap de creada, per tant això comportarà a fer una cerca.

En el cas que es vagi a fer una gestió de l'empresa i ja hi hagi una empresa creada llavors només se'n podrà fer una modificació d'aquesta creada, això per tant representarà fer una cerca, per si hi ha una empresa creada.

Les classes de disseny identificades són:

- Empresa: classe que encapsula la informació referent a una empresa.
- Entitat: superclasse que conté l'empresa

### **Gestió clients.**



Quan s'afegeix un client no caldrà fer cap comprovació prèvia, es podrà entrar el client sense cap restricció.

En el cas de la modificació d'un client caldrà fer una cerca per obtenir el client a modificar, un cop trobat és passarà a fer les pertinents modificacions.

Tant en afegir com en modificar un client tenim la opció d'assignar una zona i un agent comercial, en el cas tant l'agent comercial com la zona hauran d'existir en el sistema i per tant prèviament se'n haurà de fer una cerca per tal de comprovar que existeixin i finalment assignar en el client.

En fer una consulta del client caldrà fer la mateixa cerca que en la modificació però no se li donarà el pas a la modificació, si no que només es mostrarà.

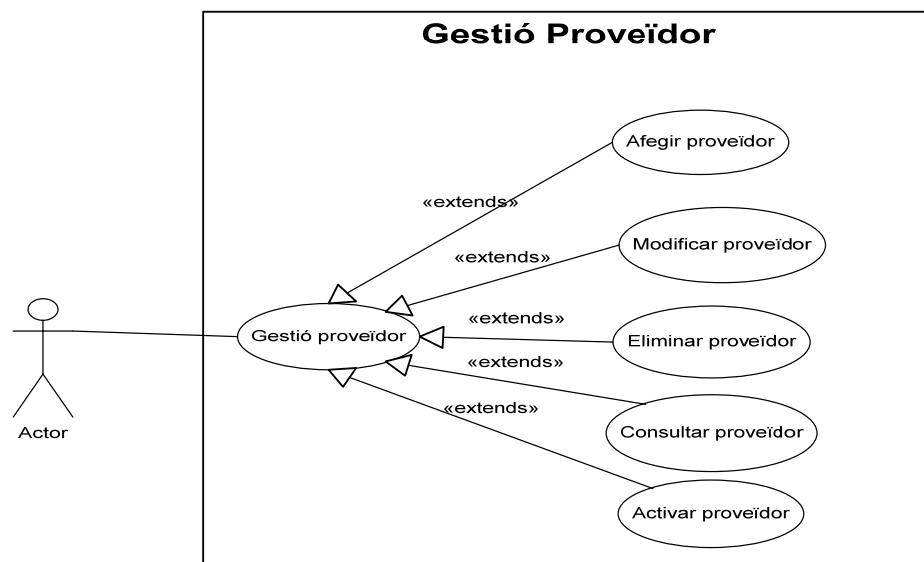
En eliminar un client primer de tot es buscarà el client i aquest no s'eliminarà del sistema només es canviarà el seu estat d'actiu a no actiu i es conservaran les seves relacions amb zona i treballador, per tenir-les guardades en cas de canviar el seu estat de no actiu a actiu en un futur.

Activar un client s'haurà de tenir en compte les mateixes passes que l'eliminar però l'estat del client és al revés.

Les classes de disseny identificades són:

- Entitat: superclasse d'EntitatComercial
- EntitatComercial: superclasse de client.
- Client: classe que encapsula les dades referents a client
- Zona: classe que encapsula les dades que fan referència a zona.
- Treballador: classe que encapsula les dades referents a treballadors.

### **Gestió proveïdor.**



Per afegir un proveïdor no hi haurà cap tipus de restricció.

En voler fer la modificació o la consulta d'un proveïdor prèviament se'n haurà de fer una cerca del proveïdor i en cas d'existir en la modificació se'n podrà fer les

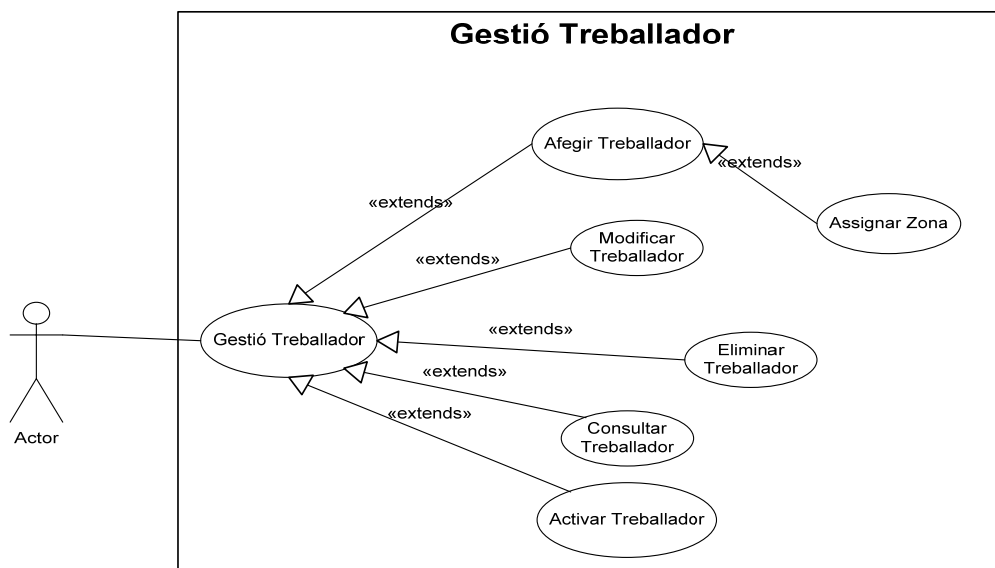
pertinents modificacions i en cas de la consulta es mostrarà el proveïdor amb la seva respectiva informació.

Tant en l'eliminació com en l'activació d'un proveïdor primer de tot s'haurà de fer una cerca del proveïdor per tal comprova que existeix en el sistema, en aquest cas si s'està eliminant aquest no desapareix del sistema si no que es canvia el seu estat a no actiu, per poder més tard activar-lo.

Les classes de disseny identificades són:

- Entitat: superclasse d'EntitatComercial
- EntitatComercial: superclasse de client.
- Proveïdor: classe que encapsula les dades referents a proveïdor

### **Gestió Treballador.**



Per afegir un treballador no caldrà fer cap comprovació prèvia. En l'entrada del client es podrà assignar una zona per tant caldrà que la zona existeixi en el sistema.

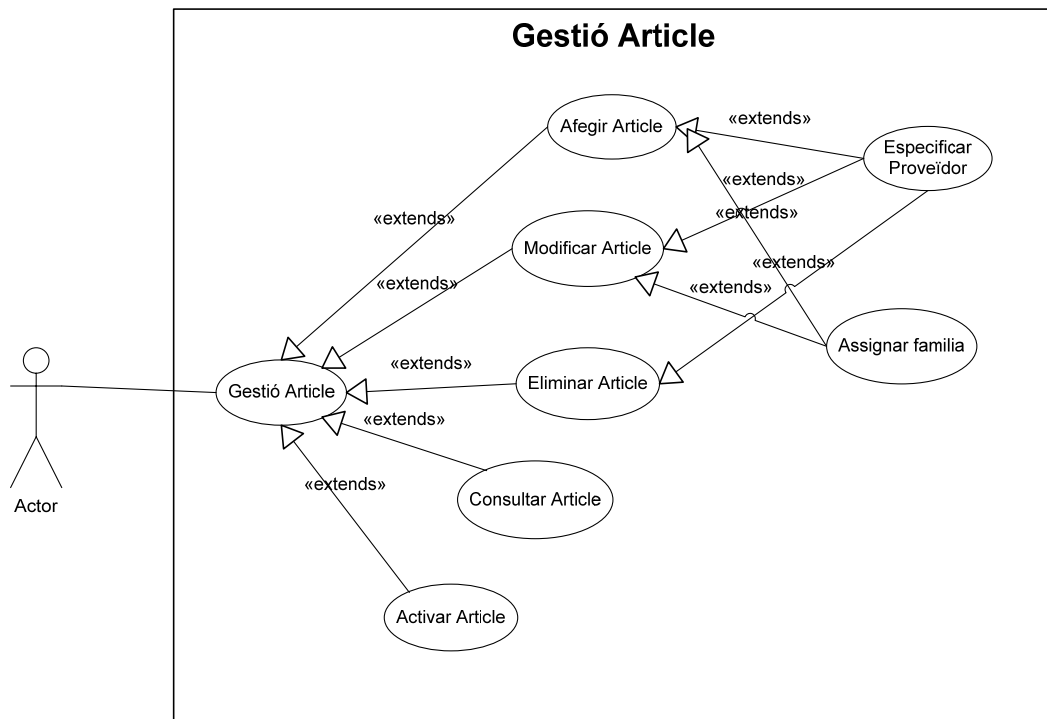
En el cas de la modificació i la consulta del treballador primer de tot caldrà fer una cerca del treballador si existeix en el sistema i tot seguit en el cas de fer la modificació es veuran les dades a modificar i en cas de la consulta es visualitzaran les dades però no se'n podrà fer cap modificació

Eliminar o activar un treballador serà necessari fer-ne una cerca per obtenir el treballador i poder-lo eliminar o activa. En eliminar un treballador si que s'eliminaran també les assignacions que té amb la zona i es canviarà el seu estat a inactiu. En el moment d'activar novament un treballador es canviarà el seu estat a actiu i en cas de voler fer noves assignacions s'haurà de fer amb una modificació.

Les classes de disseny identificades són:

- Entitat: superclasse d'EntitatComercial
- EntitatComercial: superclasse de client.
- Zona: classe que encapsula les dades que fan referència a zona.
- Treballador: classe que encapsula les dades referents a treballadors.
- Associacio\_ZonaTreballador: classe que guarda les associacions entre un Treballador i una zona.

### Gestió d'articles.



Per afegir un article no caldrà fer cap comprovació.

En la modificacions, eliminar , consultar i activar s'haurà de fer una cerca de l'article prèvia abans de procedir a fer qualsevol de les operacions.

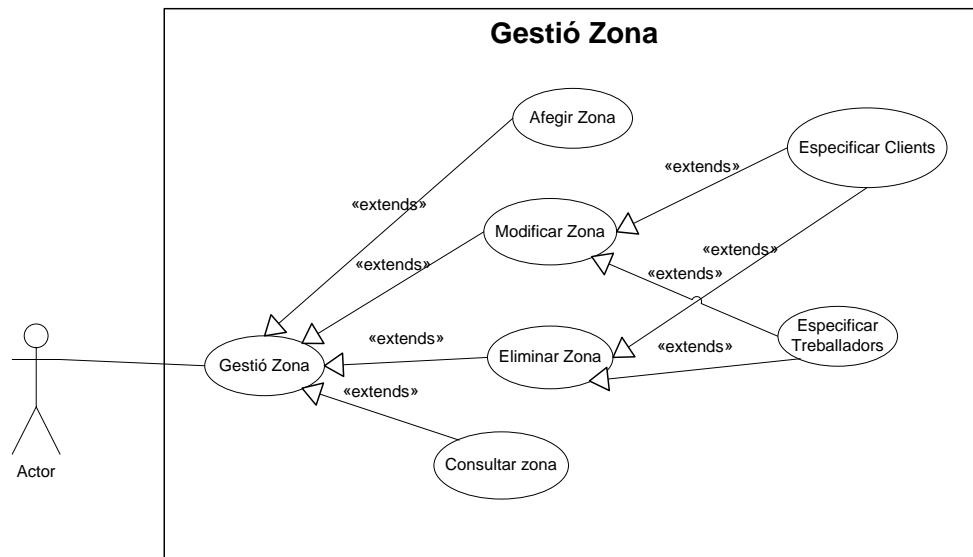
En el cas d'afegir i modificar un article s'haurà de tenir en compte que se li pot assignar una família, per tant s'haurà de fer una cerca prèvia per comprova que existeix la família. En el cas d'assignar proveïdors a un article aquests proveïdors hauran d'existir dins el sistema, de la mateixa manera quan s'indiqui quin dels proveïdor és l'habitual. En el cas de desassignar un proveïdor s'haurà de comprovar que aquest no sigui el proveïdor habitual, ja que no es coherent eliminar un proveïdor que és l'habitual de l'article.

En el cas d'eliminar un article a part de la cerca de l'article i eliminar-lo del sistema, s'hauran de fer les desassignacions, es a dir, eliminar totes les relacions que té amb un proveïdor. En eliminar l'article s'elimina directament la relació amb article, ja que es article qui guarda la relació amb família

En activar un proveïdor se'n canviarà l'estat, per poder fer de nou les assignacions s'haurà de fer des de l'operació de modificació.

Les classes de disseny identificades són:

- Article: classe que encapsula les dades referents a l'article.
- Treballador: classe que encapsula les dades referents a proveïdor, per defecte també s'usen les superclasses d'aquesta.
- Família: classe que encapsula les dades referents a al família
- Associacio\_ProveidorArticle: classe que encapsula la informació referent a la relació entre un proveïdor i un article.

**Gestió zones.**

En afegir una zona no farà falta cap comprovació prèvia, només farà falta entrar les dades necessàries.

En la modificació de zona primer s'haurà de fer una cerca de la zona, per poder comprovar que existeix, dins les modificacions que es poden fer també es podrà fer l'assignació de treballadors i clients a la zona. El que cal comprovar és:

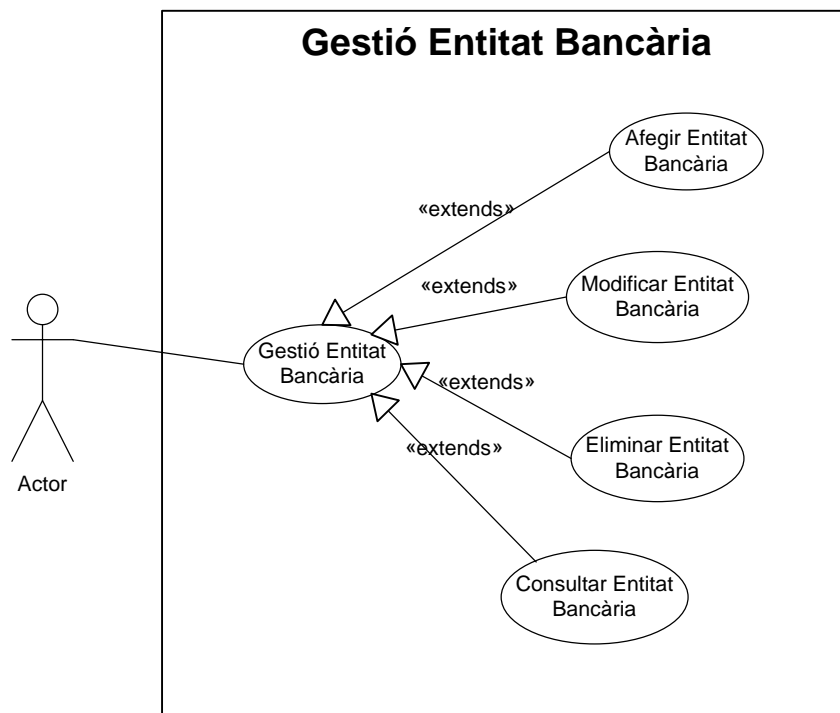
- Assignació i dessassignació de clients:
  - Assignació → cercar el client i comprovar que no té cap zona assignada i crear la relació amb la zona
  - Dessassignació → cercar el client, comprovar que la zona que té assignada és la que s'està modificant i trencar la relació.
- Dessassignar i Assignar treballadors:
  - Dessassignar → cercar el treballador i comprovar que existeix i comprovar que hi ha una relació amb la zona, en cas que existeixi, eliminar la relació.
  - Assignar → cercar el treballador, comprovar que no hi ha cap relació amb la zona a modificar i crear la relació

En eliminar una zona s'haurà de primer buscar que la zona existeixi, i fer totes les desassignacions amb els treballadors i els clients com s'ha explicat en l'operació de modificació i tot seguit eliminar la zona.

Les classes de disseny identificades són:

- zona: classe que encapsula tota la informació referent a la zona
- Treballador: classe que encapsula la informació referent a treballador.
- Client: classe que encapsula la informació del client.
- Associació\_ZonaTreballador: classe que encapsula la informació referent a la relació entre zona i treballador.

### **Gestió Entitat Bancària.**



Les classes de disseny identificades són:

En el moment d'afegir, modificar, consultar o eliminar una entitat bancària primer de tot s'haurà de fer una cerca de l'entitat bancària per comprovar que existeixi i tot seguit les operacions, ja esmentades.

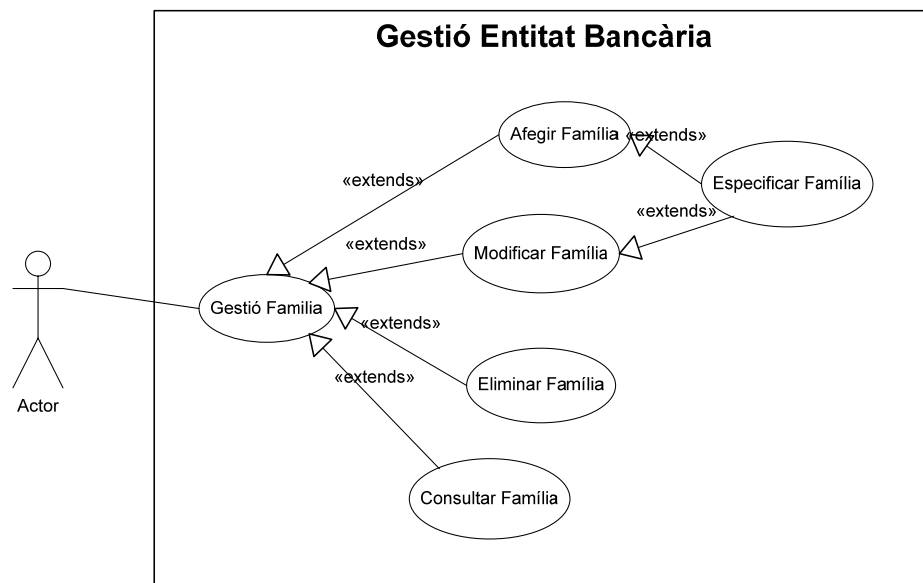


Si en voler afegir una entitat aquesta ja existeix l'aplicació haurà d'avisar de l'existència i per tant no es podrà afegir de nou.

En fer una modificació després de la cerca se'n podrà fer les modificacions pertinents.

L'entitat bancària es podrà eliminar sense cap tipus de restricció.

### **Gestió de famílies.**



En el moment de crear una nova família, afegir-la, no caldrà fer cap comprovació prèvia, només caldrà que s'entri la informació de la família i si forma part d'una altre família relacionar-la amb la família, per això caldrà que aquesta família existeixi en el sistema, per tant se'n haurà de fer una cerca.

En el cas de les modificacions també haurem de fer una cerca sobre la família a modificar i canviar la informació necessària, en aquest cas si es vol fe una modificació de la família pertanyent es podrà fer sempre hi quan aquesta no tingui subfamílies relacionades amb ella.

En eliminar una família s'haurà de comprovar que aquesta no tingui subfamílies relacionades, en cas de tenir-ne no es podrà eliminar i s'hi no en té aquesta passarà a no formar part del sistema, per tant s'esborrarà i també s'eliminarà la relació amb la família superior.

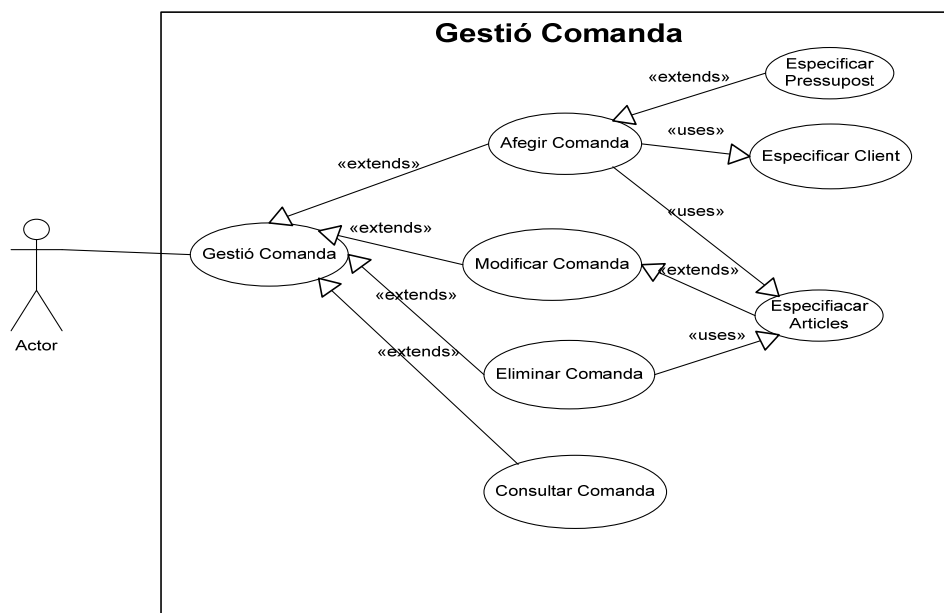
En la consulta només farà falta una cerca sobre la família a consultar.

Les classes de disseny identificades són:

- Família: classe que encapsula la informació referent a família.

## Gestió de vendes

### Generació de comandes de clients



En voler afegir una comanda, si aquesta es fa directament sense la utilització d'un pressupost no s'haurà de fer cap més comprovació que, el client a qui vagi dirigida la comanda i els articles que s'hi afegeixin existeixin dins el sistema, es a dir, que en caldrà fer una cerca, en afegir-los caldrà fer les relacions pertinents de cadascun d'ells amb la comanda. En cas de generar la comanda a partir d'un pressupost aquesta se li assignarà automàticament tots els articles i el client a qui anava dirigit el pressupost i no es podrà fer cap modificació.

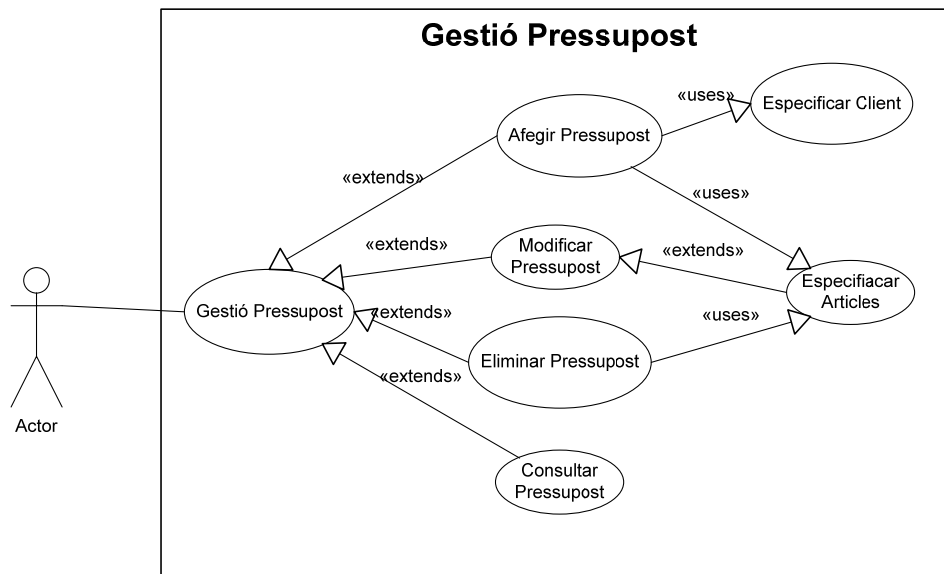
En voler fer una modificació de la comanda primer de tot caldrà fer una cerca de la comanda a modificar. En cas d'existir dins el sistema caldrà comprovar cap document l'utilitzi per la seva creació i que ella no utilitzi cap document per haver-se generat, es a dir, que no tingui cap relació amb cap document. En cas de

poder modificar la comanda no es podrà canviar a quin client anava dirigida, però si els articles, en cas d'afegir articles caldrà crear les noves relacions i en cas contrari d'eliminar-los s'haurà d'eliminar la relació amb la comanda.

En el cas d'eliminar una comanda, primer de tot caldrà cercar la comanda i comprovar que cap document l'utilitzi per la seva generació, en cas que si, no es podrà fer la modificació, altrament si no té aquest tipus de relació amb cap document, és podrà eliminar la comanda trencant tota relació amb els article (de manera que caldrà modificar-ne l'estoc), trencar la relació amb el client i finalment si té una relació amb un document (pressupost) per que la comanda s'ha generat a partir d'aquest, es trencarà la relació i el document amb el que estava relacionat canviarà el seu estat a aprovat, per tant tornarà estar actiu.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- DocumentClient: superclasse que hereta de Document que encapsula la informació de documents de clients.
- PressupostVenda: classe que hereta de DocumentClient que encapsula la informació referent als pressupostos de Clients.
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class que encapsula la informació d'articles.

**Realitzar pressupost**

Alhora d'afegir un pressupost s'haurà de tenir en comptes que sempre haurà d'estar lligat a un client i a una conjunt d'articles. Cal que el client i cada un dels articles formin part del sistema, es per això que s'haurà de fer una cerca per cadascun d'ells.

En el moment d'afegir articles al pressupost caldrà tenir present de que no s'afegeixin articles que el seu estoc no cobreixi la quantitat entrada en el pressupost. Un cop s'afegeixi el pressupost s'hauran de crear les relacions pertinents amb article i client. Cal esmentar que tota la informació de totals les calcularà internament el programa.

En la modificació del pressupost només es podrà fer en el cas que aquest no s'hagi donat per realitzat, per tant en el moment de fer la cerca del pressupost si aquest esta realitzat no s'haurà de poder permetre, altrament per modificar-lo només es podrà fer sobre els articles que hi estan relacionats i les dates, per conseqüència la informació de totals també podrà quedar modificada (segons els articles)

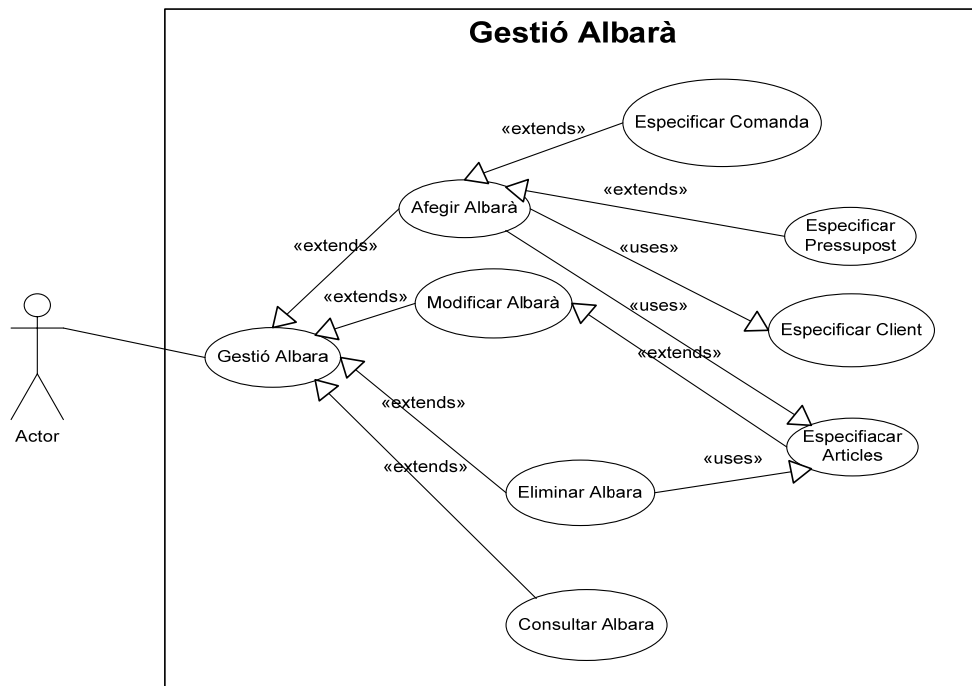
Per eliminar un pressupost caldrà fer una cerca del pressupost per obtenir-lo i mirar que aquest no estigui relacionat amb cap d'altre document, es a dir, no

s'hagi donat per aprovat. En cas de poder eliminar-lo s'hauran d'eliminar prèviament les relacions amb els articles i el client.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- DocumentClient: superclasse que hereta de Document que encapsula la informació de documents de clients.
- PressupostVenta: classe que hereta de DocumentClient que encapsula la informació referent als pressupostos de Clients.
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class que encapsula la informació d'articles.

### **Gestió de lliurament de producte o servei.**

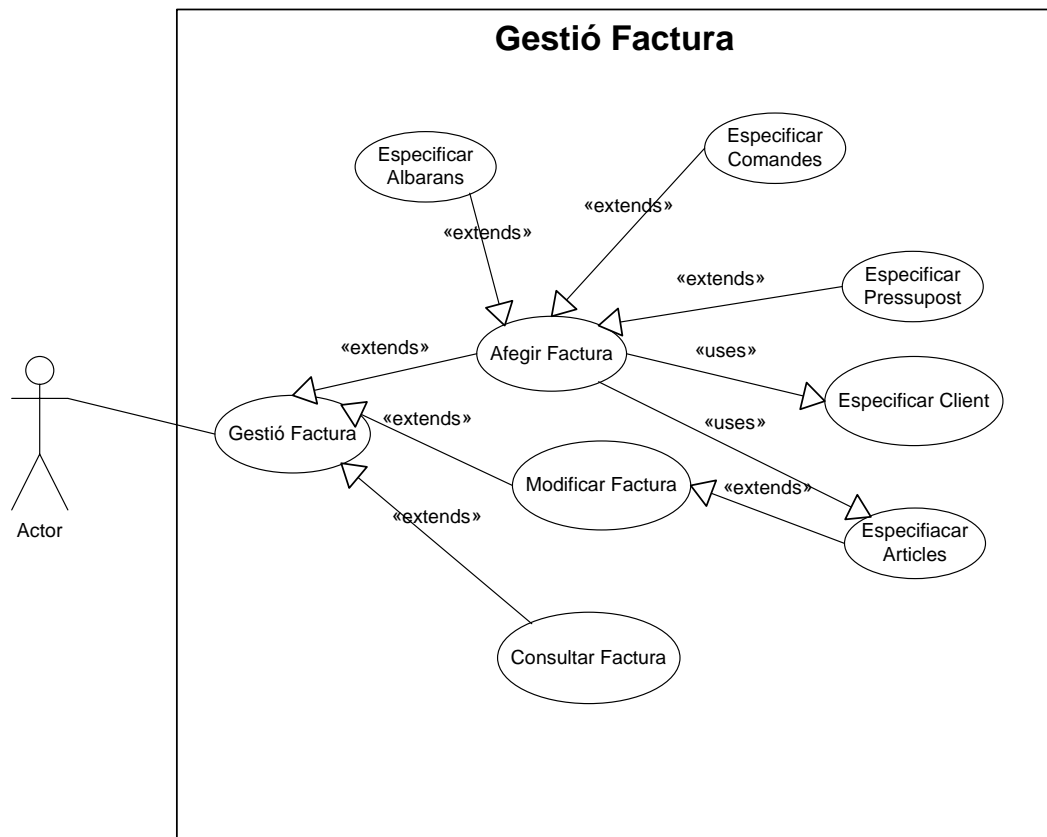


La gestió de lliurament de producte el casos d'ús es exactament igual que la generació de la comanda, però s'afegeix que un albarà es pot crear a partir d'una comanda. L'eliminar, modificar i consultar tindran les mateixes especificacions que comanda

En el cas d'afegir a partir de comanda, es podrà realitzar fent d'una selecció de totes les comandes existents que estiguin relacionades amb rang de clients i per el seu rang de dates, en aquest cas s'haurà de fer una cerca de totes les comandes "pendents", es a dir no realitzades (que cap altre document les utilitzi) dins d'unes dates d'entrega, que estiguin relacionades amb un rang de clients. El resultat d'aquesta cerca es crearà un albarà per cada una de les comandes, on per cada albarà creat tindrà una relació amb la comanda assignada i un client i per conseqüència de la relació amb comanda també estarà relacionat indirectament amb un conjunt d'articles (que seran els que estan relacionats amb la comanda). La única data entrar serà la data nova d'entrega amb les que s'hauran de generar els albarans

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- DocumentClient: superclasse que hereta de Document que encapsula la informació de documents de clients.
- PressupostVenta: classe que hereta de DocumentClient que encapsula la informació referent als pressupostos de Clients.
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class eque encapsula la informació d'articles.
- AlbaràVenta: classe que encapsula tota la informació referent als albarans.
- ComandaVenta: classe que encapsula tota la informació referent als albarans

**Gestió de facturació**

La gestió de facturació té el mateix cas d'ús que la gestió d'albarans amb la peculiaritat que una factura es pot generar a partir de molt albarans i que una factura no es podrà eliminar, per requisits que s'han donat.

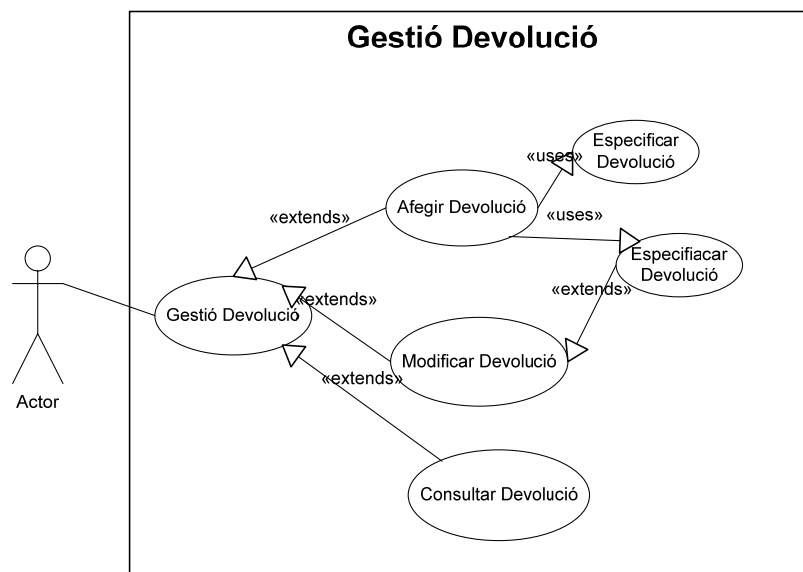
Les operacions de modificació i consulta es faran amb la mateixa metodologia que amb la comanda i albarà.

Així que a l'hora d'afegir factures que fan referència d'albarans també es farà com en el cas de la creació d'albarans a partir de comandes, de manera que primer caldrà fer una selecció de tots els albarans que es troben amb el rang dates entrat i que corresponguin al rang de clients. Per tant cada un dels albarans existirà dins el sistema, per tant és de sobre entén que si existeix un albarà, hi haurà un client relacionat amb ell existent. La diferència que hi ha amb la generació d'albarans a partir de comanda és que una factura fa referència a molt albarans.

- Document: superclasse que fa referència als documents

- PressupostVenda: classe que hereta de Document que encapsula la informació referent als pressupostos de Clients.
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class eque encapsula la informació d'articles.
- AlbaràVenda: classe que encapsula tota la informació referent als albarans.
- ComandaVenda: classe que encapsula tota la informació referent als albarans
- FacturaVenda: classe que encapsula tota la informació referent a les factures

### Generació de devolució



Alhora d'afegir una devolució s'haurà de tenir en comptes que sempre haurà d'estar lligat a un client i a una conjunt d'articles. Cal que el client i cada un dels articles formin part del sistema, es per això que s'haurà de fer una cerca per cadascun d'ells.

En el moment d'afegir articles a la devolució caldrà tenir present de no s'afegir articles que el seu estoc no cobreixi la quantitat entrada en el pressupost. Un cop s'afegeixi la devolució s'hauran de crear les relacions pertinents amb article i



client. Cal esmentar que tota la informació de totals les calcularà internament el programa.

En la modificació de la devolució només es podrà fer sobre els articles que hi estan relacionats i les dates, per conseqüència la informació de totals també podrà quedar modificada (segons els articles), en el moment que una devolució se li hagi cobrat per tant estarà en un estat de realitzada no es podrà modificar.

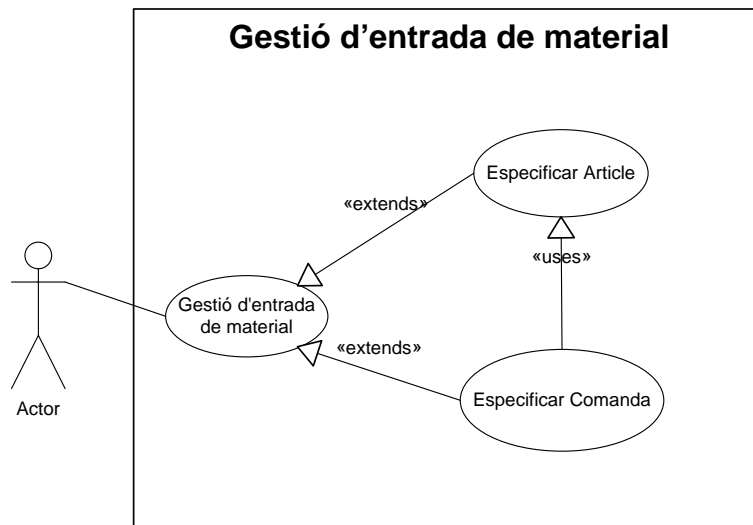
Per eliminar una devolució caldrà fer una cerca de la devolució per obtenir-la i mirar que aquest no estigui relacionat amb cap d'altre document, es a dir, no s'hagi donat per realitzat. En cas de poder eliminar-la s'hauran d'eliminar prèviament les relacions amb els articles i el client.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class eque encapsula la informació d'articles.
- FacturaVenta: classe que encapsula tota la informació referent a les factures

## Gestió de compres

### Gestió d'entrada de material



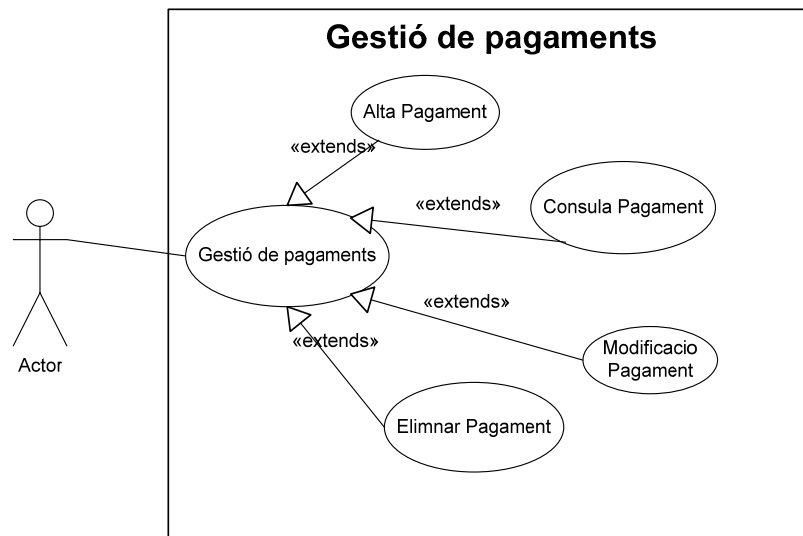
Per l'entrada de material es podrà fer a partir d'un article o a partir d'una comanda de compra i afegir tots els articles de la comanda.

Per fer l'entrada a partir d'un article, primer caldrà fer una cerca per tal de validar que l'article existeix en el sistema i només caldrà augmentar el nivell d'estoc que entrarà l'actor per pantalla.

Si l'entrada d'articles es fa a partir de comanda de compra s'haurà de cercar la comanda amb la qual es vol actualitzar el nivell d'estoc d'articles i comprovar que existeix només caldrà validar-la i tot seguit obtenir tots els articles que estan relacionats amb ella i augmentar l'estoc segons indica la relació.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- Comanda Compra: classe que hereta de document i encapsula tota la informació sobre les comandes de compra.
- Article: classe que encapsula tota la informació d'articles.

**Gestió de pagaments**

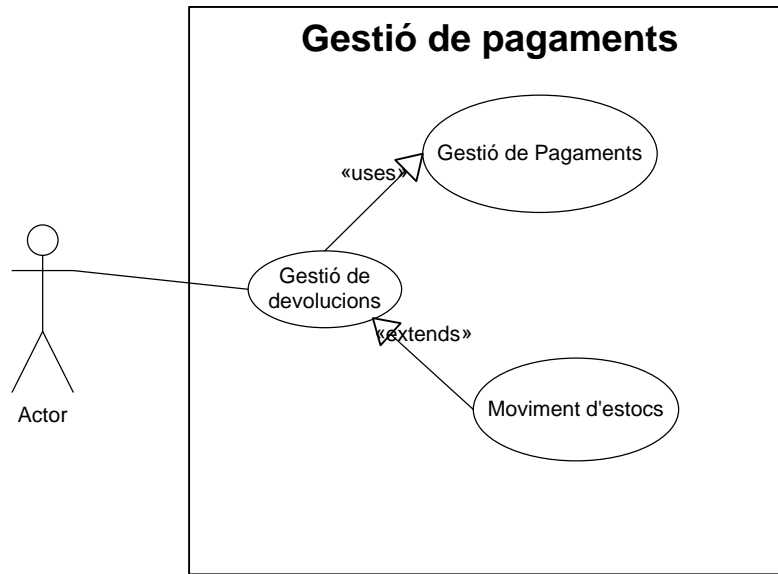
Per afegir un pagament primer de tot s'haurà de fer la cerca a quin proveïdor fa referència el document a pagar, tot seguit, caldrà comprovar que no s'hagi entrat ja aquell document en el sistema, es a dir, una cerca per comprovar la no existència del document i tot seguit es podrà acabar d'entrar la informació referent al pagament.

Per fer la modificació caldrà buscar quin es el document a modificar, si el document existeix s'obtindrà i sen podrà fer la modificació necessària, de tal manera que només es podrà fer la modificació de la data de pagament que es la que valida que aquell document ja esta realitzat.

Per eliminar una pagament primer de tot s'haurà de buscar aquest pagament i si aquest no està pagat es podrà eliminar en cas contrari, no.

Les classes de disseny identificades són:

- Factura compra: classe que encapsula les dades de les factures que arriben del proveïdor.

**Gestió de devolucions**

Aquest cas d'ús usa el cas d'ús anterior, seguirà la mateixa metodologia i pot incloure ús de la gestió de moviments que s'explicarà seguidament.

Les classes de disseny identificades són:

- Factura compra: classe que encapsula les dades de les factures que arriben del proveïdor.
- Article: classe que encapsula la informació referent a l'article.

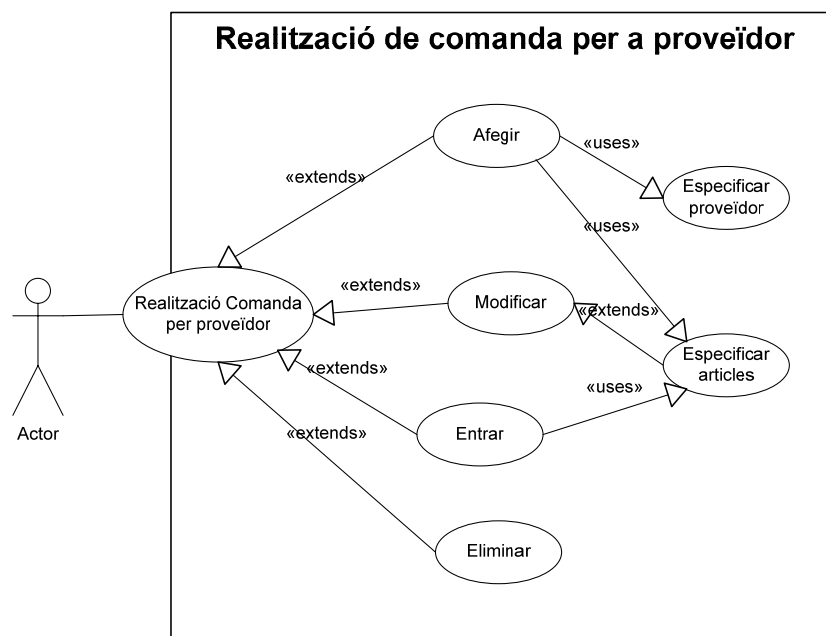
**Control d'estocs de magatzem****Realització de comandes per a proveïdors**

Afegir una comanda comporta que ha d'estar relacionada amb un proveïdor que haurà d'existir a la base de dades, per tant se'n farà una cerca per comprova que existeix. I també haurà de tenir un o molts articles relacionats per tant cadascun d'ells haurà d'existir en el sistema i més hauran d'estar relacionats amb el proveïdor que estarà relacionat amb la comanda a entrar. En realitzar la comanda no s'haurà de fer cap control d'estoc.

Modificar la comanda, nomes es podrà fer si la comanda existeix en el sistema i aquesta no ha estat validada.

Entrar (Validar) per poder validar-la primer de tot haurem de cercar la comanda i un cop verificat que existeix en el sistema es passarà a veure les relacions que té amb article i per cada una se'n farà el corresponent control d'estoc.

Per eliminar una comanda primer de tot s'haurà de cercar i tot seguit si la comanda no s'ha validat es podrà eliminar, altrament no.



Les classes de disseny identificades són:

- ComandaCompra: classe que encapsula tota la informació referent a la comanda d'una compra
- Article: classe que encapsula la informació referent a l'article.
- Proveïdor: classe que encapsula la informació referent al proveïdor
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.

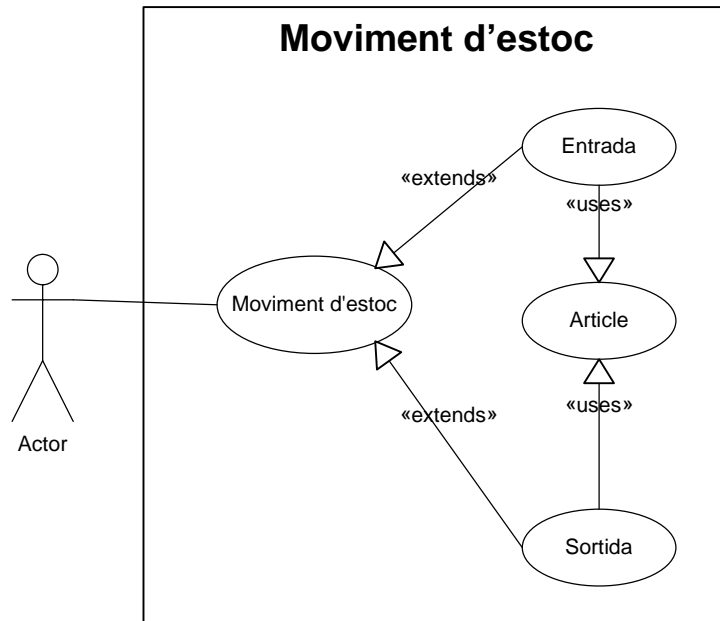
### **Consultes d'estoc baix**

La consulta d'estoc baix, només caldrà fer una cerca de tots els articles que hi ha al sistema i mostrar-los en una llista.

Les classes de disseny identificades són:

- Article: classe que encapsula la informació referent a l'article.

### **Moviment d'estoc**



En el moviment d'estoc t'han es podrà fer un moviment d'entrada com un de sortida, en els dos casos caldrà primer de tot cercar l'article del qual sen vol fer el moviment i tot segui sen farà la modificació d'estoc, segons la quantitat que l'actor entri per pantalla, en el cas d'una entrada augmentarà i en cas de la sortida disminuirà la quantitat d'estoc.

Les classes de disseny identificades són:

- Article: classe que encapsula la informació referent a l'article.

## ***Resum d'ingressos i Despeses i Consultes***

### ***Consulta sobre Clients***

En la consulta sobre clients primer de tot s'haurà d'obtenir mitjançant una cerca el client triat a fer la consulta i tot seguit es farà buscarà totes les relacions amb factures que se li han entregat.

Per cada una de les factures es buscaran els articles que hi ha relacionats i es filtraran els 10 del qual se'n hagin venut més.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- Client: classe que encapsula la informació de clients.
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: class eque encapsula la informació d'articles.
- FacturaVenta: classe que encapsula tota la informació referent a les factures

### ***Consulta sobre Proveïdors***

Es buscarà el proveïdor del qual es vol fer la consulta, si aquest existeix a la base de dades se'n buscaran totes les relacions que té amb les factures de compres es crearà una llista per les pagades i una altre per les impagades i es mostraran per pantalla.

Les classes de disseny identificades són:

- FacturaCompra: classe que encapsula tota la informació referent a les factures.
- Proveïdor: classe que encapsula tota la informació referent al proveïdor.

**Consulta sobre articles**

Es farà una selecció de tots els articles que hi ha en el sistema i amb les relacions amb les factures generades a clients es traurà un llistat del total que se n'ha venut. Per altre banda es farà una altre selecció de tots els articles existents i amb les relacions amb les comandes de compra es farà un filtre per saber la quantitat total que se n'ha comprat de cada un, que s'ha fet

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- LiniaDocument: classe que encapsula la informació de cadascuna de les línies del document.
- Article: classe que encapsula la informació d'articles.
- FacturaVenta: classe que encapsula tota la informació referent a les factures

**Consulta d'Ingressos i despeses**

Per mostrar els ingressos es farà una consulta que de selecció de totes les factures de venda donades per cobrades i en farà la suma dels imports, per les despeses farà una selecció de les factures que s'han pagat als proveïdors i també en farà la suma.

Les classes de disseny identificades són:

- Document: superclasse que fa referència als documents
- Article: classe que encapsula la informació d'articles.
- FacturaVenta: classe que encapsula tota la informació referent a les factures
- FacturaCompra: classe que encapsula tota la informació referent a les factures.



## 4.4 Disseny de classes

En aquest apartat es transforma el model de classes lògic, provinent de l'anàlisi en un model de classes de disseny, el qual ha ha l'especificació detallada de les classes, els seus atributs, operacions i les relacions entre elles.

Els tipus d'atributs i paràmetres es mostren en notació del llenguatge de programació Java, com el que s'ha utilitzat en la construcció del projecte.

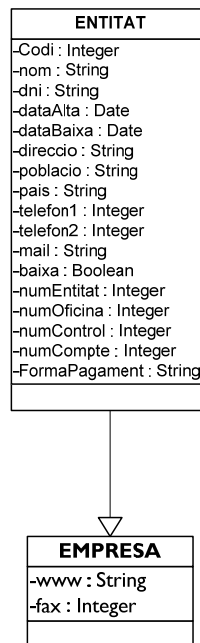
En el dibuix de les classes no es mostren les operacions per simplificar el dibuix (ja que quedaven massa grans), però en el detall de cada una de les classes, ja s'especifica cada una de les operacions.

Cal recordar que en el disseny de classes no s'hi introdueixen aquelles classes pròpies del llenguatge de desenvolupament com ara classes relacionades amb les interfícies d'usuari o aquelles que fan referència a la gestió de llistes d'elements, que també son pròpies de les eines de desenvolupament.

Una de les altres coses ha tenir en compte que degut a l'explorabilitat entre classes hi ha implícits atributs per tal de saber amb quin objecte estan relacionats. En l'estudi de les classes de disseny, no s'inclou cap d'aquets atribut, ja que se sobre entén que hi són mitjançant les associacions dels diagrames que es mostren

## Gestió de dades físiques

### Gestió de la informació de l'empresa



### Entitat

#### Atributs

Codi: identificador de l'entitat.

Nom : nom de la persona a qui fa referència.

Dni: identificadador de la persona o empresa a nivell nacional.

Data Alta: data del dia que va entrar a l'empresa.

Data Baixa: data del dia que es va donar de baixa.

Direcció: direcció fiscal de l'empresa.

Població: que pertany l'empresa o persona física.

País: país que pertany l'empresa o persona física.

Telefon1: primer telèfon.

Telefon2: segon telèfon.

Mail: correu electrònic de l'entitat.

Baixa: estat en que es troba dins el sistema l'entitat

numEntitat: número de l'entitat bancaria del conta corrent.

numOficina: número de l'oficina e l'entitat bancaria que forma part.

Control: numero de control del contate corrent.

numCompte: numero de compte de la conta corrent

### Operacions

Entitat() → constructor per defecte.

Entitat(int codiEntitat, int codiBanc, String nom, String dni, Date dataAlta, Date dataBaixa, String direccio, String poblacio, String pais, int telefon1, int telefon2, String mail, boolean baixa, int numEntitat, int numOficina, int control, int numCompte, String FormaPagament)

→ Constructor amb paràmeters de la classe.

Més totes les operacions necessàries per entrar i obtenir els diferents atributs

## **Empresa**

### Atributs

www: direcció web de l'empresa

fax: número del telèfon fax.

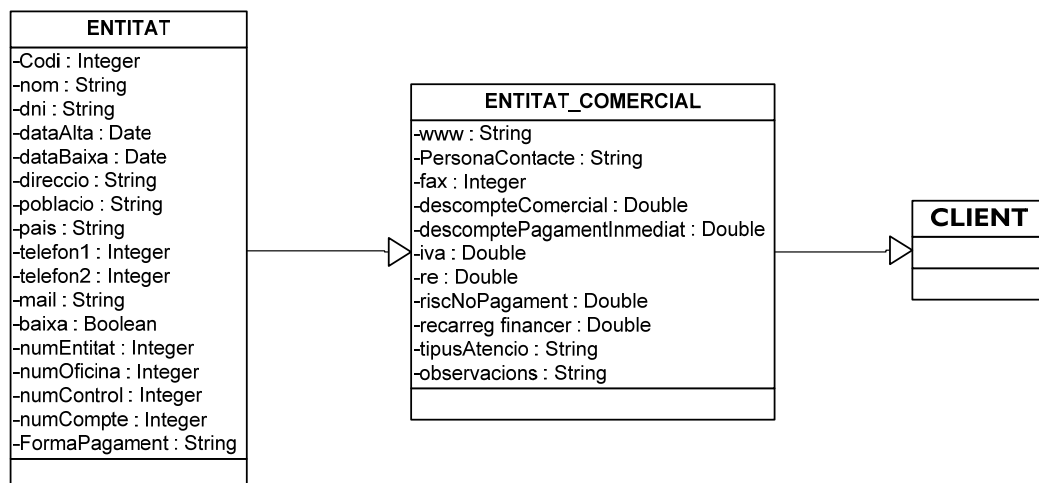
A més tindrà tots els atributs que hereti de Entitat

Operacions

Heretara totes les operacions d'Entitat i redefinirà el constructor per paràmetres, amb la inclusió dels dos paràmetres de més

Empresa () → Constructor per defecte

Empresa(String www, int fax) → constructor amb els dos parametres propis d'empresa.

**Gestió dels clients**

Entitat → ja explicada

**Entitat\_Comercial**Atributs

www: direcció de la pàgina web de l'entitat comercial

Persona Contacte: nom de la persona a posar-se en contacte amb l'empresa

Fax: número de telèfon fax

descompteComercial: valor de descompte comercial, en tant per cent.

descomPagamentInmediat: valor del descompte en cas de ser immediat, en tant per cent

Iva: valor de l'iva que s'aplica a l'entitat, en tant per cent

Re: recàrrec d'equivalència en tant per cent.

recarregFinancer: valor en tant per cent del recàrrec financer que s'aplica a l'entitat.

Observacions: altre informació que cal ser necessària de l'entitat comercial.

### Operacions

EntitatComercial() → constructor per defecte;

EntitatComercial(int codiEntitat, int codiBanc, String nom, String dni, Date dataAlta, Date dataBaixa, String direccio, String poblacio, String pais, int telefon1, int telefon2, String mail, boolean baixa, int numEntitat, int numOficina, int control, String numCompte, String FormaPagament)

→ Constructor amb tots els paràmetres que pot tenir una entitat comercial

EntitatComercial(int codiEntitatComercial, String nomEntitatComercial, String www, String personaContacte, int fax, double descompteComercial, double descomptePagamentInmediat, double iva, double re, double riscNoPagament, double recarregFinancer, double irpf, String observacions)

→ constructor amb els paràmetres propis de l'entitat comercial

## **Client**

### Atributs

Tarifa: guardarà quina es la tarifa que se li aplicarà al client (no es mostrarà en el diagrama)

La resta d'atributs els hereta d'entitat i entitat comercial

Operacions

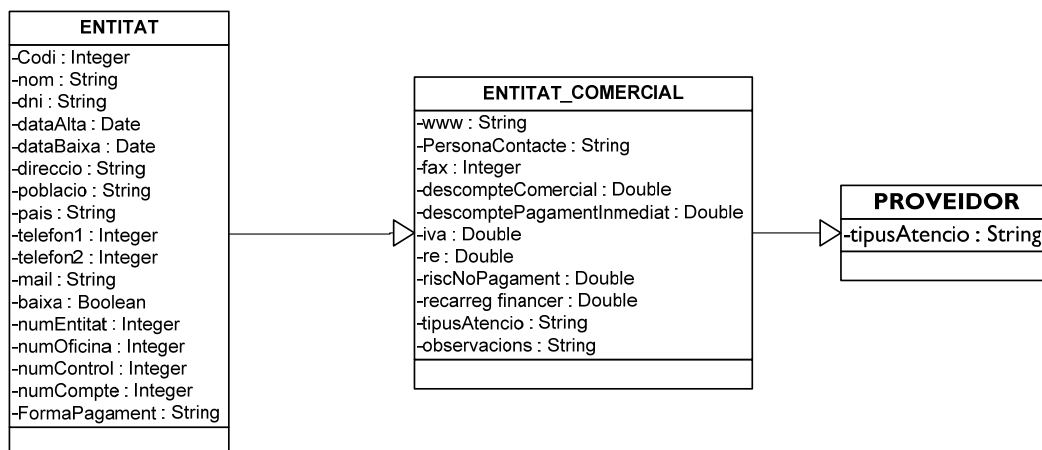
Client() → constructor per defecte.

Client(int codiClient, int codiZona, String tarifa, int codiTreballador)

→ constructor amb els paràmetres únics de client

Client(int codiEntitat, int codiBanc, String nom, String dni, Date dataAlta, Date dataBaixa, String direccio, String poblacio, String pais, int telefon1, int telefon2, String mail, boolean baixa, int numEntitat, int numOficina, int control, String numCompte, String FormaPagament)

→ constructor amb tots el paràmetres que pugui tenir, inclosos els que hereta

**Gestió de proveïdors**

Entitat i Entitat comercial → ja estan explicades

**Proveïdor**Atributs

A part de tots els que hereta d'entitat i entitat\_comercial tindrà

tipusAtenció: guarda una cadena que fa referència a (Sr, Sra, Srta,)

### Operacions

Proveïdor() → constructor per defecte

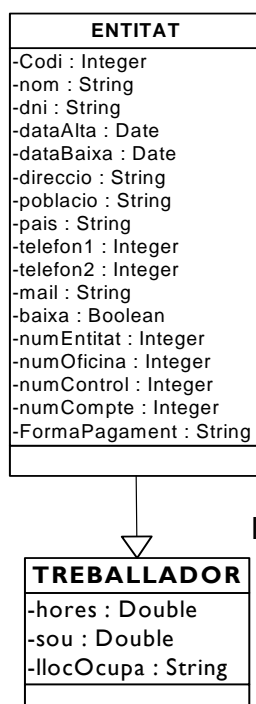
Proveïdor(int codiEntitat, int codiBanc, String nom, String dni, Date dataAlta, Date dataBaixa, String direccio, String poblacio, String pais, int telefon1, int telefon2, String mail, boolean baixa, int numEntitat, int numOficina, int control, int numCompte, String FormaPagament, int codiEntitatComercial, String nomEntitatComercial, String www, String personaContacte, int fax, double descompteComercial, double descomptePagamentInmediat, double iva, double re, double riscNoPagament, double recarregFinancer, String observacions, String tipusAtencio)

→ constructor amb tots els atributs que pot tenir proveïdor.

Proveïdor (int codiProveïdor, String tipusAtencio)

→ constructor amb els atributs únic de proveïdor

### **Gestió treballador**



Entitat → ja explicada

### **Treballador**

#### Atributs

Hores: hores que treballa segons la forma de pagament

Sou: sou que cobra segons la forma de pagament

Lloc Ocupa: quin és la seva situació en les tasques de l'empresa

#### Operacions

Treballador()

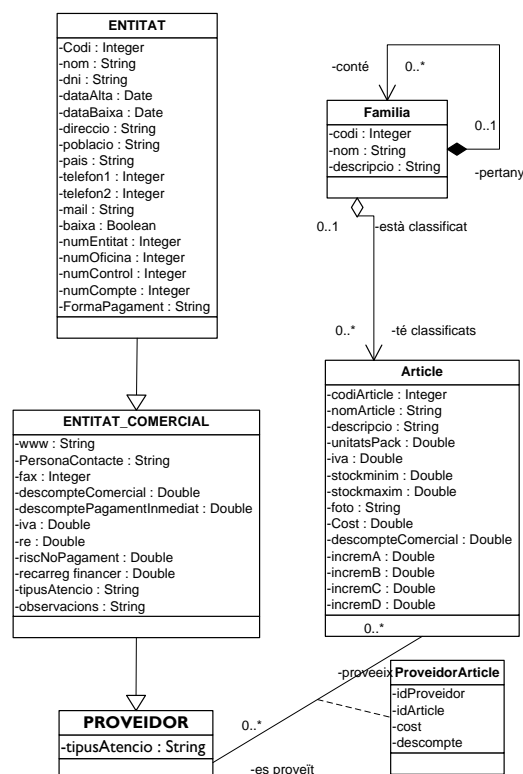
→ constructor per defecte

Treballador(int codiEntitat, int codiBanc, String nom, String dni, Date dataAlta, Date dataBaixa, String direccio, String poblacio, String pais, int telefon1, int telefon2, String mail, boolean baixa, int numEntitat, int numOficina, int control, int numCompte, String FormaPagament, int codiTreballador, double hores, double sou)

→ constructor amb tots els atributs que pot tenir.

Treballador(int codiTreballador, double hores, double sou, String llocOcupa)

## Gestió d'articles



Entitat, Entitat\_Comercial i Proveïdor → ja explicades.



**Article**

Atributs (alguns dels atributs no es mostren en el dibuix, ja que finalment s'han canviat per els següents)

codiArticle; codi identificatiu de l'article.

nomDescripcio; nom descriptiu de l'article.

descripció; descripció detallada que ha de tenir l'article

imatge; imatge de l'article

iva; impost que s'aplica.

stockMax; estoc màxim que hi ha cabuda en el magatzem

stockMin; estoc mínim a tenir en compte per fer comandes

uni\_caixa; unitats que es troben a cada caixa.

cost; cost que té l'article comprat al proveïdor

costnet; cost total que ens costa l'article.

marge1; increment percentual que s'aplica al cost net per obtenir el preu definitiu en la tarifa A

marge2; increment percentual que s'aplica al cost net per obtenir el preu definitiu en la tarifa B

marge3; increment percentual que s'aplica al cost net per obtenir el preu definitiu en la tarifa C

marge4; increment percentual que s'aplica al cost net per obtenir el preu definitiu en la tarifa D

preu1; Preu sense iva de la tarifa A

preu2; Preu sense iva de la tarifa B

preu3; Preu sense iva de la tarifa C

preu4; Preu sense iva de la tarifa D

quantitat: Estoc que es troba en el magatzem

descompte: descompte aplicat per le proveedor.

Baixa: esta en que es troba l'article dins el programa.

### Operacions

Article() → constructor per defecte

Article(int codiArticle, String nomDescripcio, String descripcio, String imatge, Double iva, int codiFamilia, double stockMax, double stockMin, double uni\_caixa, int codiProveedorHabitual, double cost, double costnet, double marge1, double marge2, double marge3, double marge4, double preu1, double preu2, double preu3, double preu4, double quantitat, double descompte, boolean baixa)

→ constructor amb els paràmetres necessaris per crear un article.

### **ProveedorArticle**

Classe resultant de l'associació entre Article i Proveïdor

### Atributs

codiArticleProveedor: codi identificatiu de l'associació

cost: preu que té el producte provinent del proveïdor.

Descompte. Descompte que el proveïdor ens fa sobre el producte en la venta

cost\_net: cost final que té el producte segons el proveïdor.

### Operacions

ProveedorArticle → constructor per defecte.

ProveedorArticle(int codiArticleProveedor, Article a, Proveedor p, double cost, double descompte, double cost\_net)

→ constructor amb les paràmetres necessaris.

## Familia

codiFamilia: codi que tindrà la família.

NomFamilia: nom que defineix la família.

DescripcióFamília: breu descripció de les característiques de la mateixa.

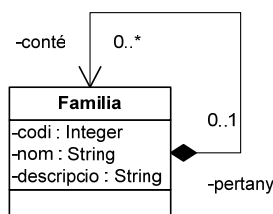
## Operacions

Familia() → constructor per defecte

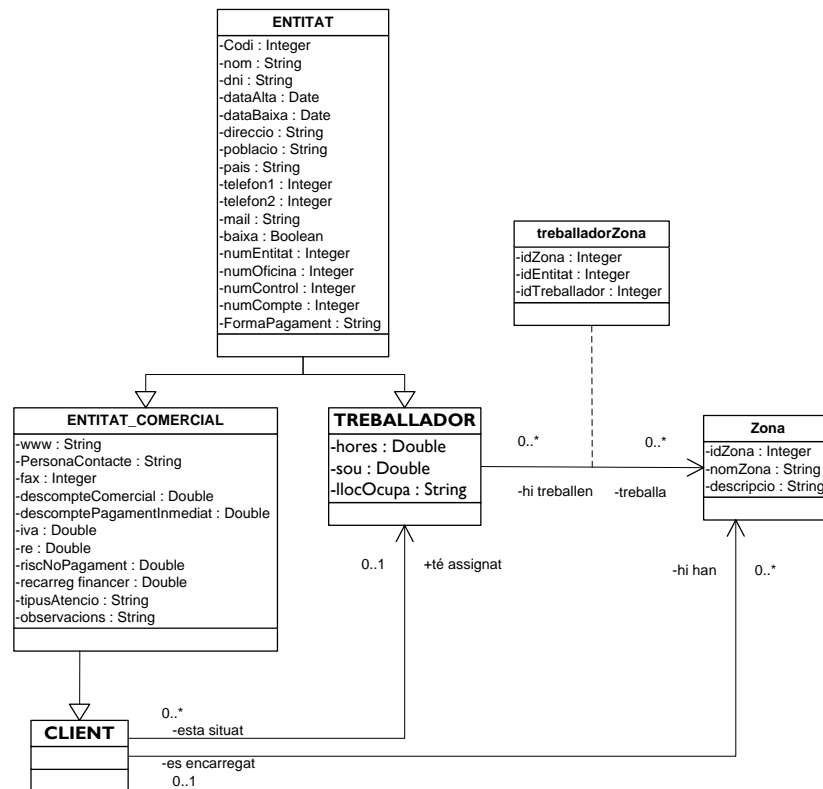
Familia(int codiZona, String nomFamilia, String descripcioFamilia

→ constructor mitjançant paràmetres.

## Gestió famílies



Família → ja explicada en la gestió d'articles

**Gestió de les zones**

Entitat, Entitat\_Comercial, Client, Treballador → classes ja explicades anteriorment

**Zona**Atributs

codiZona: codi que identifica la zona.

nomZona: nom que tindrà la zona

descripcioZona; descripció breu de les característiques de la zona

Operacions

Zona() → constructor per defecte

Zona(int codiZona, String nomZona, String descripcioZona)

→ constructor amb els pertinents paràmetres necessaris per la creació de zona

### ***Gestió d'entitats bancària***

#### Atributs

codiBanc: codi amb el que s'identificarà el banc.

numEntitat: número amb el que s'identifica l'entitat bancària.

numOficina: número amb el que s'identifica l'oficina de l'entitat bancària.

nomEntitat: nom de l'entitat bancària.

nomOficina: nom de l'oficina de l'entitat.

Domicile: direcció on és troba l'oficina.

Poblacio: població on es troba l'oficina.

Pais: país on s'ubica la oficina

personaContacte: nom de la persona amb qui ens podem posar en contacte amb l'oficina.

telefon1: primera opció de telèfon.

telefon2: segona opció de telèfon

fax: número del telèfon fax.

#### Operacions

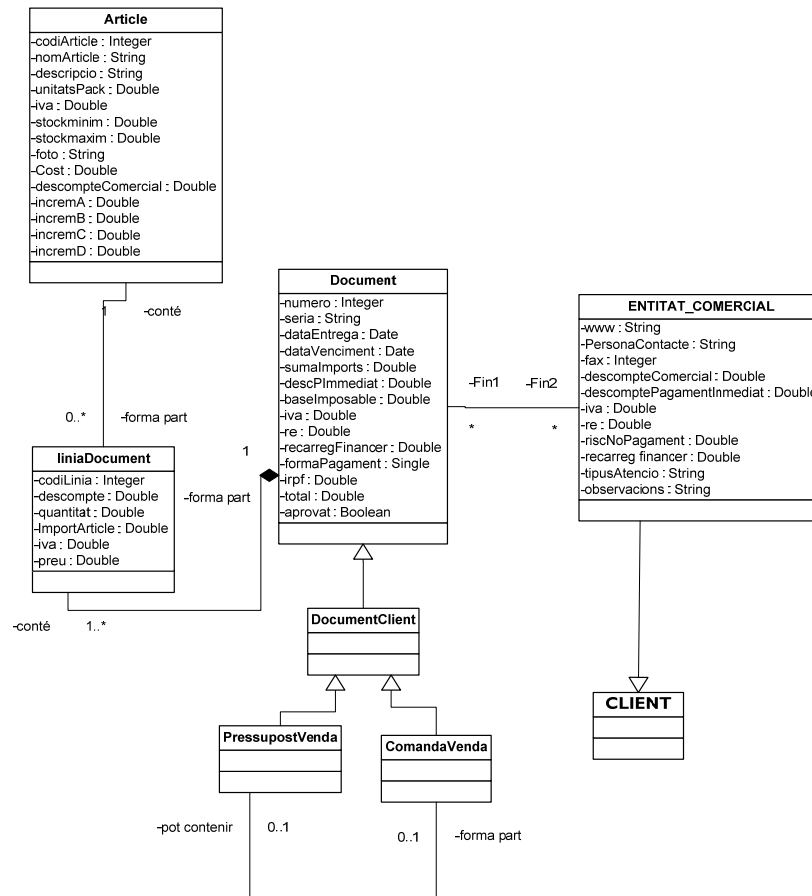
EntitatBancaria() → constructor per defecte

EntitatBancaria(int codiBanc, int numEntitat, int numOficina, int control, int numCompte, String nomEntitat, String nomOficina, String domicili, String poblacio, String pais, String personaContacte, int telefon1, int telefon2, int fax)

→ constructor amb els parameters necessaries per a la construcció una entitat bancària.

#### 4.4.2 Gestió de vendes

##### Generació de comandes de clients



Entitat\_Comercial, Article i Client → Classes ja explicades anteriorment.

#### Document

##### Atributs

codiDocument: identificador del document.

numDocument: numero que tindrà el document, per identificar-lo

seria: caràcters que identificarà el tipus de document.

dataEntrega: data que és farà l'entrega del document

dataVenciment: data de caducitat del document.

sumalImports: suma total de la factura sense calcular els impostos

desPlmmediat: és el valor a que tenen els descomptes sobre la sumalImports

baseImposable: és el resultat de descomptar els descomptes a la sumalImports

iva: guarda el valor que té iva sobre la base imposable.

Re: guarda el valor que té re sobre la base imposable.

Recarregfinancer: valor que té el recàrrec financer sobre la base imposable.

Irp: guarda el valor de l'irpf del document sobre la base imposable

Total: guarda el total a pagar del document

Aprovat: estat en que es troba el document

### Operacions

Document() → constructor per defecte

Document(int codiDocument, int numDocument, String seria,vEntitatComercial titular, Date dataEntrega, Date dataVenciment,vdouble sumalImports, double descPlmmediat, double baseImposable,vdouble iva, double re, double recarregFinancer,vString formaPagament, double irpf, double total, boolean realitzat,vList<liniaDocument> liniaDoc)

→ constructor amb els parametres necessaris per la creació d'un Document

### liniaDocument.

#### Atributs

codiLinia: codi identificador de la linia.

Preu: cost que tindrà l'article en aquella línia (ja que pot ser modificable al preu de l'article)

Descompte: descompte aplicat al preu de l'article

Quantitat: quantitat d'articles representa la venda.

ImportArt: guarda l'import total de l'article, fent els descompte corresponent i la multiplicación per la quantitat.

Iva: valor d'iva aplicat.

### Operacions

liniaDocument() → constructor per defecte de la línia

liniaDocument(int codiLinia, Article article, double preu, double descompte, double quantitat, double importArt, double iva, Document doc)

→ constructor de la línia amb els paràmetres necessaris per la seva creació.

## **DocumentClient**

### Atributs

Els que hereta de Document

### Operacions

DocumentClient() → constructor per defecte.

DocumentClient(Document DocumentAssociat) → constructor del document amb el paràmetre d'associació.

DocumentsClient(int codiDocument, int numDocument, String seria, EntitatComercial titular, Date dataEntrega, Date dataVenciment, double sumalImports, double descPlImmediat, double baseImposable, double iva, double



re, double recarregFinancer, String formaPagament, double irpf, double total, boolean realitzat, List<liniaDocument> liniaDoc)

→ constructor amb els parameters necessaries per la construcció.

## PressupostVenda

Els atributs seran heretats directament de les classes superiors i també les operacions.

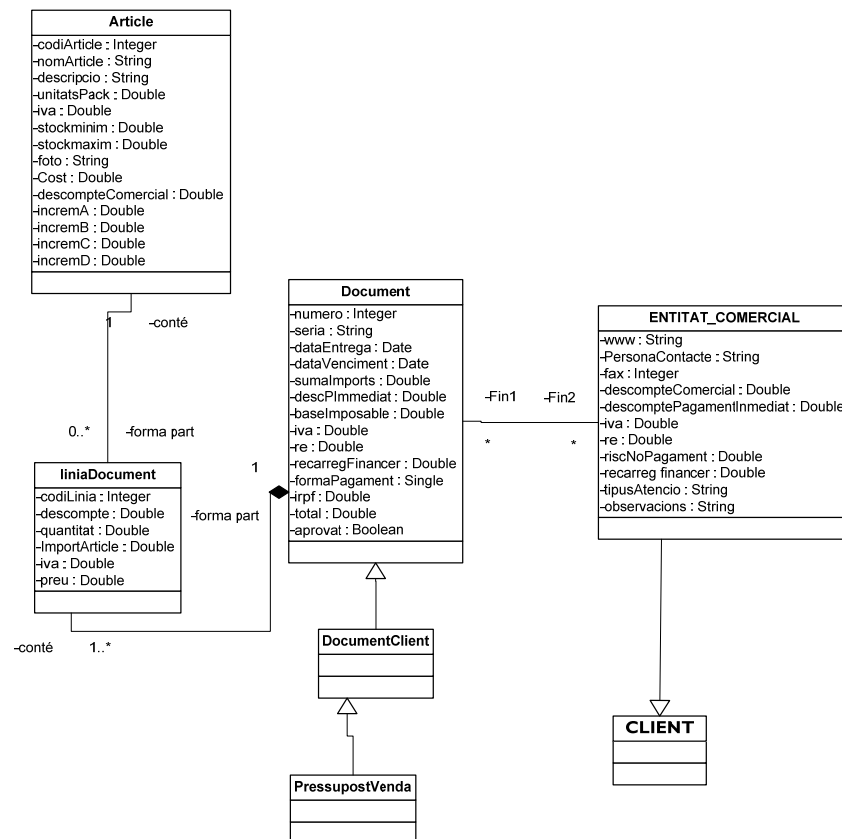
PressupostVenda() → constructor per defecte que dóna valor VP a la seria

## ComandaVenda

Atributs i operacions heretades per les superclasses

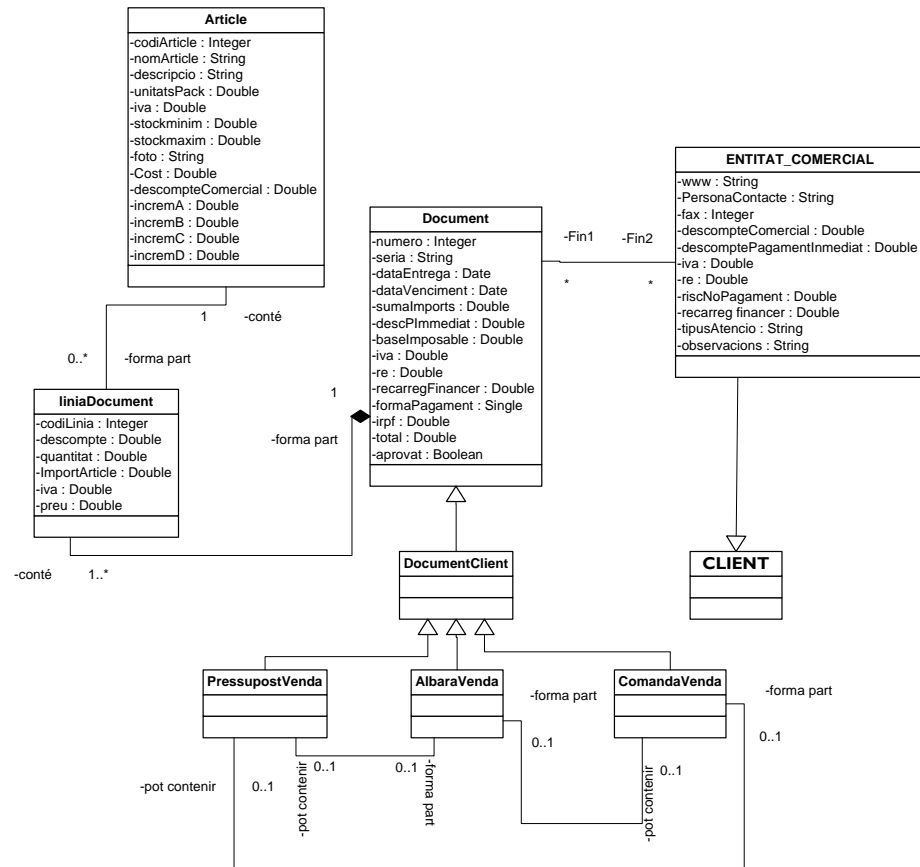
ComandaVenda() → constructor per defecte que dóna valor CP a la seria

## Realització de pressupostos



Totes les classes estan explicades en els anteriors apartats.

### Gestió de lliurament de producte o servei



Les classes que intervenen són: Article, liniaDocument, Document, DocumentClient, PressupostVenda, AlbaraVenda, EntitatComercial, Client, ComandaVenda.

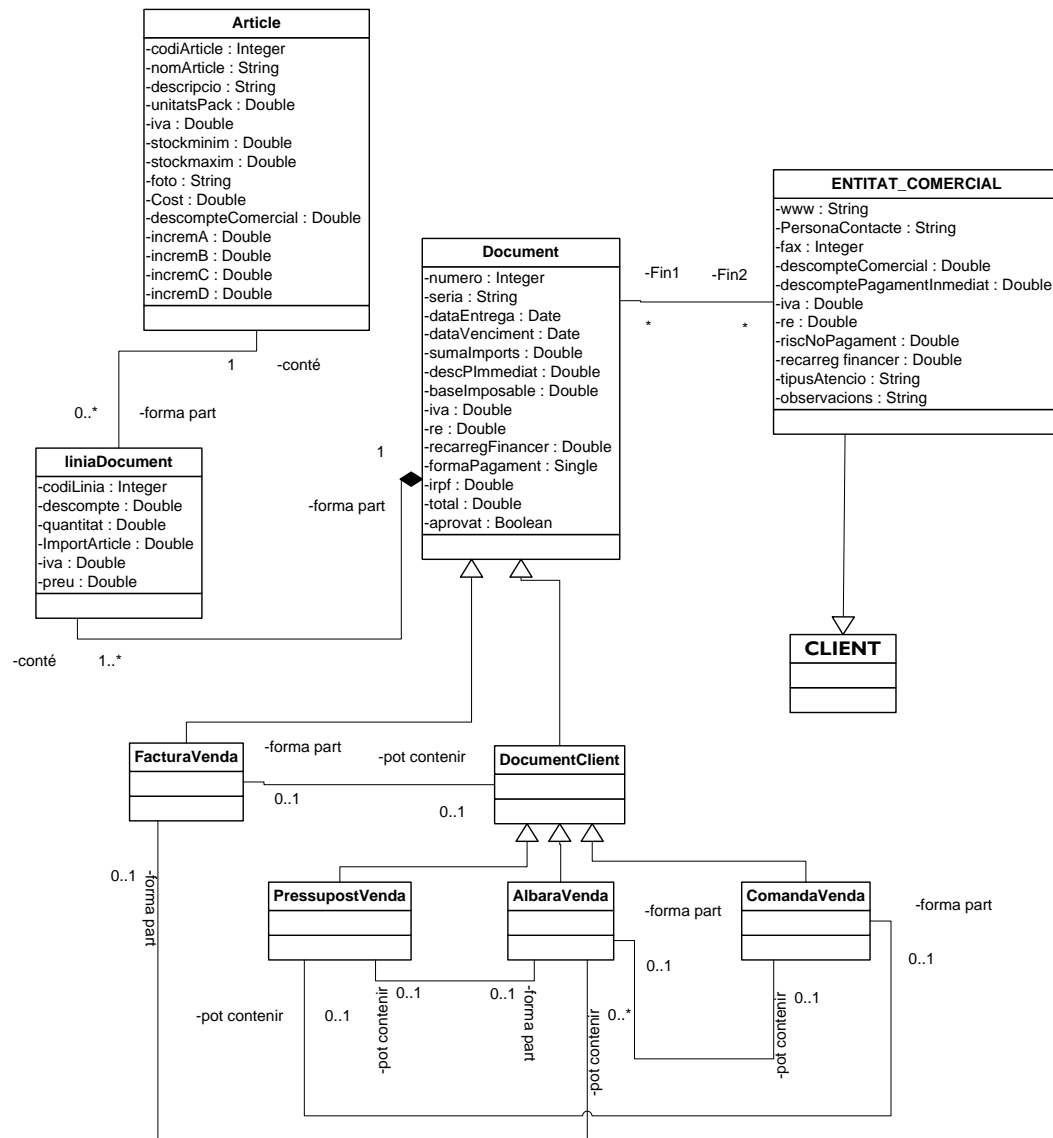
La classe no explicada encara es ComandaVenda.

### AlbaraVenda

Els atributs i les operacions son les heretades de les classes superiors DocumentClient i Document.

### Operacions

AlbaraVenda() → constructor per defecte que inicialitza la seria a VA

**Gestió de cobrament**

Les classes que intervenen són: Article, liniaDocument, Document, DocumentClient, PressupostVenda, AlbaraVenda, EntitatComercial, Client, ComandaVenda i FacturaVenda.

**FacturaVenda**

Hereta els atributs de la Superclasse Document i les seves operacions

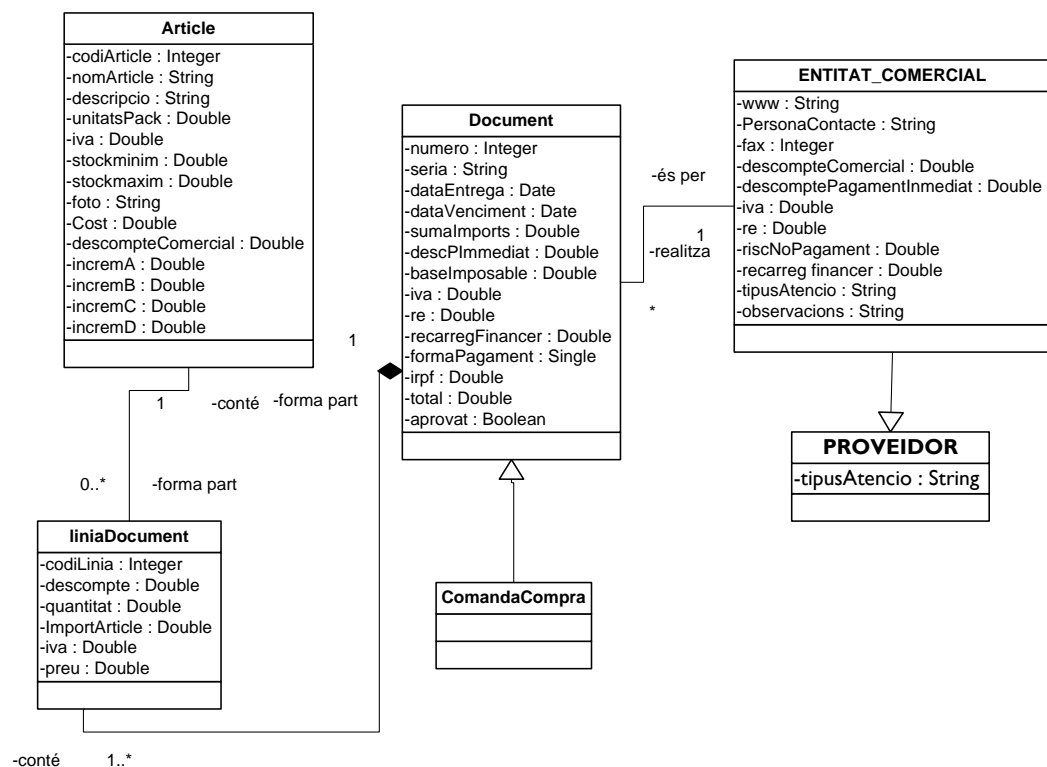
Operacions

FacturaVenda() → constructor per defecte que inicialitza la seria amb el valor VF.

FacturaVenda(List<DocumentsClient> clients) → constructor per paràmetres.

**Gestió de devolució**

El diagrama utilitzat i les classes són exactament les mateixes que la gestió de cobrament.

**Gestió de compres****Gestió d'entrada de material**

Les classes implicades en aquest subsistema són Article, liniaDocument, Document i ComandaCompra. Totes excepte ComandaCompra estan explicades anteriorment.

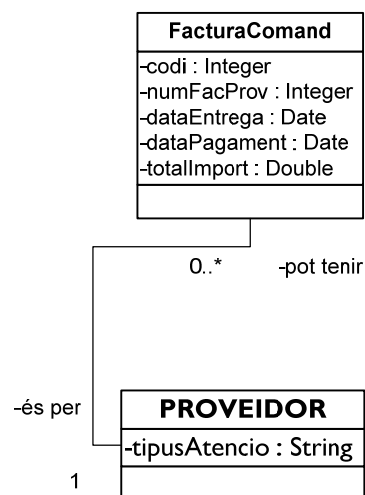
## ComandaCompra

Hereda de Document els atributs i les operacions.

### Operació

ComandaCompra() → constructor per defecte que inicialitza la seria a CC.

## Gestió de pagaments



Proveïdor i FacturaCompra són les dues classes que intervenen en aquest sistema on Proveïdor, ja s'ha explicat.

## FacturaComanda

Codi: Identificador de l'objecte FacturaComanda

numFacProv: numero que té la factura del proveïdor.

dataEntrega: data que porta el document que prové del proveïdor.

dataPagament: data de quan es va fer el pagament de la factura

totalImport: total a pagar al proveïdor

## Operacions

FacturaCompra() → constructor per defecte.

FacturaCompra(int codi, String numFacProv, Date dataEntrega, Date dataPagament, double totallImport, Proveidor proveidor)

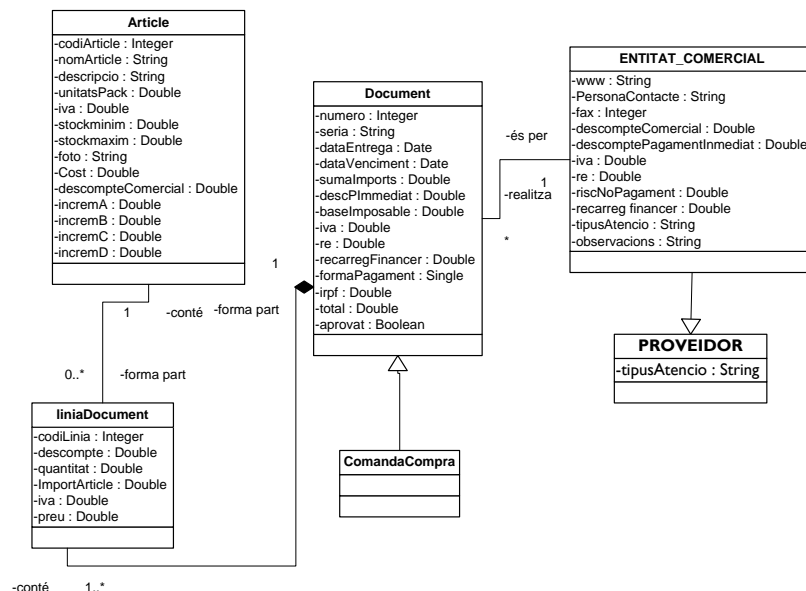
→ constructor de l'objecte factura amb els paràmetres necessaris.

## Gestió de devolucions

Aquest subsistema tindrà el mateix diagrama de classes que Gestió de pagaments, per tant també les mateixes classes associades.

## Control d'estocs de magatzem

## Realització de comandes per a proveïdors



Totes les classes implicades ja estan explicades en els apartats anteriors

**Consultes d'estoc baix**

Article
-codiArticle : Integer
-nomArticle : String
-descripcio : String
-unitatsPack : Double
-iva : Double
-stock : Double
-stockminim : Double
-stockmaxim : Double
-foto : String
-Cost : Double
-descompteComercial : Double
-incrementA : Double
-incrementB : Double
-incrementC : Double
-incrementD : Double

La única classe implicada dins aquest subcas és l'article que es el que guarda el total d'estoc.

**Gestió d'estocs**

Article
-codiArticle : Integer
-nomArticle : String
-descripcio : String
-unitatsPack : Double
-iva : Double
-stock : Double
-stockminim : Double
-stockmaxim : Double
-foto : String
-Cost : Double
-descompteComercial : Double
-incrementA : Double
-incrementB : Double
-incrementC : Double
-incrementD : Double

Com en el cas anterior la gestió d'estoc d'entrada i sortida també serà l'article la única classe implicada.

**Resum d'ingressos i despeses i Consultes**

En aquest cas d'us es tornen a veure totes les classes que s'han anat veient al llarg d'aquest capítol i són consultes que engloben gairebé totes les classes que intervenen en la de gestió de compres

## 4.5 Disseny físic de dades

En aquest apartat es realitza el disseny físic de la base de dades del sistema, a partir dels models de classes de disseny del sistema.

### ***4.5.1 Transformació del model de classes al model físic***

Abans de fer la transformació del model de classe al model de dades físiques s'haurà de tenir en compte que en el model de classes teniem dos tipus d'herència.

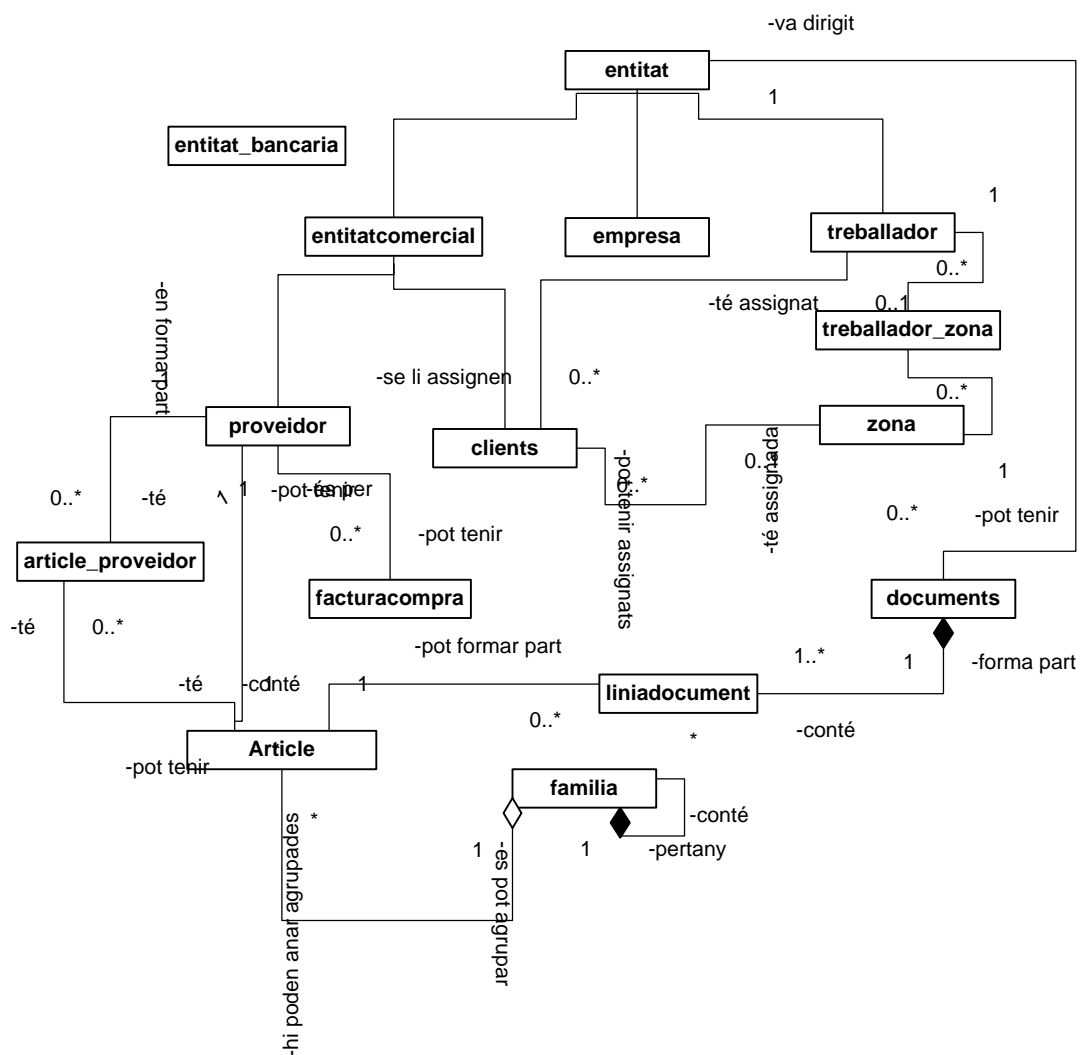
- Un que englobava les persones o empreses físiques, es a dir, les classes Entitat, EntitatComercial, Client, Proveïdor i Treballador
- Una segona herència que feia referència a diferents tipus de documents.

En el primer cas d'un estudi de com seria la millor manera de guardar les dades s'ha arribat a la conclusió que per cada una de les classes serà una taula a la base de dades que s'enllaçaran a partir de les claus foranes.

En el segon cas s'ha pensat que la millor manera de guardar els diferents documents es a partir d'una sola taula, ja que, de totes maneres cadascún dels tipus entre ells es diferenciarien per la seria.

Tot seguit ja podem fer mostra del diagrama relacional de la base de dades.



























































Tot seguit s'explicarà cada taula quins camps té, quin dels camps es clau primària, quin són clau forana si el seu valor pot ser nul o no.

















**Taula Article**

En aquesta taula com el seu nom indica s'emmagatzemarà la informació referent al articles. La clau primària en aquest cas és el codi de client i les claus foranes són el codiProveidorHabitual per la relació que hi ha amb el proveïdor (1..N) i el codi de Família per la relació (1..N) entre article i família.

Column Name	Datatype	NOT NULL
 codiArticle	 INT(5)	
 nomDescriptiu	 VARCHAR(20)	
 descripcio	 VARCHAR(200)	
 imatge	 VARCHAR(200)	
 iva	 DOUBLE	
 unitatsCaixa	 DOUBLE	
 stockMin	 DOUBLE	
 stockMax	 DOUBLE	
 codiFamília	 INT(3)	
 codiProveidorH...	 INT(3)	
 cost	 DOUBLE	
 costNet	 DOUBLE	
 marge1	 DOUBLE	
 marge2	 DOUBLE	
 marge3	 DOUBLE	
 marge4	 DOUBLE	
 preu1	 DOUBLE	
 preu2	 DOUBLE	
 preu3	 DOUBLE	
 preu4	 DOUBLE	
 descompte	 DOUBLE	
 baixa	 TINYINT(1)	
 quantitat	 DOUBLE	

***Taula article\_proveedor***

Taula resultant de la relació [N..M] entre proveïdor i article. On per agilitzar la feina d'obtenció de dades se li dona un identificado. Per tant les claus primàries són `codiArticleProveedor`, `codiArticle`, `codiEntitat`, `codiEntitatComercial` i `codiProveedor`, on a l'hora aquestes quatre últimes també són claus foranes de la taula.













Column Name	Datatype	NOT NULL
 <code>codiArticleProve...</code>	 INTEGER	✓
 <code>codiArticle</code>	 INTEGER	✓
 <code>codiProveedor</code>	 INTEGER	✓
 <code>cost</code>	 DOUBLE	✓
 <code>descompte</code>	 DOUBLE	✓
 <code>cost_net</code>	 DOUBLE	✓
 <code>codiEntitat</code>	 INTEGER	✓
 <code>codiEntitatCome...</code>	 INTEGER	✓

***Taula Clients***










































Aquesta taula consta de dues claus primàries que son `codiEntitatComercial` i `codiClient` i tres claus foranes que són `codiEntitatComercial`, `codizona` i `codiTreballador`.

La clau principal `codiEntitatComercial` és referent a l'associació d'herència que hi ha entre `EntitatComercial` i `Client`, i per aquesta raó `codiEntitatComercial` també es clau forana de client.

La clau forana `codiZona` és per la relació (1..N) que hi ha entres la tal taula client i la de zona. I finalment `codiTreballador`, és el fruit de la relació que hi ha entre client i treballador.

Column Name	Datatype	NOT NULL
 <code>codiEntitatCome...</code>	 INT(4)	✓
 <code>codiClient</code>	 INT(4)	✓
 <code>irpf</code>	 INT(2)	
 <code>codiZona</code>	 INT(3)	
 <code>tarifa</code>	 VARCHAR(2)	
 <code>codiTreballador</code>	 INT(4)	

**Taula document**









Column Name	Datatype	NOT NULL
 codi	 INTEGER	
 numDocument	 INTEGER	
 seria	 VARCHAR(3)	
 codiEntitat	 VARCHAR(45)	
 dataEntrega	 DATETIME	
 dataVenciment	 DATETIME	
 sumaImports	 DOUBLE	
 descPlImmediat	 DOUBLE	
 baseImposable	 DOUBLE	
 iva	 DOUBLE	
 recarregFinancer	 DOUBLE	
 formaPagament	 VARCHAR(45)	
 irpf	 DOUBLE	
 total	 DOUBLE	
 docAssociat	 INTEGER	
 docUtilitzat	 INTEGER	
 re	 DOUBLE	
 realitzat	 TINYINT(1)	

Aquesta taula representa tots els documents de pressupostos, comandes de venda i de compra, albarans i factures.

Per tal de poder veure les relacions entre els documents s'han afegit dos camps un és docAssociat i l'altre es docUtilitzat. El primer fa referència al document que l'ha utilitzat per la seva creació i el segon camp fa referència al document que aquest ha utilitzat per crear-ne. Per exemple per entendre-ho millor si una comanda utilitza un pressupost per realitzar-se en el cap docUtilitzat de comanda hi haurà el codi del pressupost, mentre que en el camp docAssociat del pressupost hi haurà el codi de la comanda.

La clau primària de document és el codi de document, tot hi que es podria haver optat per fer una clau primària conjunta entre seria i numDocument. Les claus foranes seran docAssociat, docUtilitzat i codiEntitat on les dues primeres, ja s'han exposat en el paràgraf anterior, codiEntitat és la clau forana que fa referència a la relació que hi ha entre documents i Entitat.







































**Taula Empresa**

Column Name	Datatype
 codiEntitat	 INTEGER
 codiEmpresa	 INTEGER
 www	 VARCHAR
 fax	 DOUBLE

Aquesta taula estarà formada per dues clau primaries codiEntitat i codiEmpresa on codiEntitat es del fruit de la relació d'herència, en aquest cas també serà la clau forana.


























### ***Taula Entitat***

Aquesta taula fa referència a la superclasse Entitat del model de classes, on la clau primària és el codi Entitat. En aquesta taula no hi ha cap clau forana.

Column Name	Datatype	NOT NULL
 codiEntitat	 INT(4)	
 nom	 VARCHAR(45)	
 nif	 VARCHAR(9)	
 dataAlta	 DATETIME	
 dataBaixa	 DATETIME	
 direccio	 VARCHAR(45)	
 poblacio	 VARCHAR(25)	
 pais	 VARCHAR(20)	
 telefon1	 INT(10)	
 telefon2	 INT(10)	
 mail	 VARCHAR(45)	
 baixa	 TINYINT(1)	
 numEntitat	 INTEGER	
 numOficina	 INTEGER	
 control	 INTEGER	
 numCompte	 VARCHAR(10)	
 FormaPagament	 VARCHAR(45)	





























### ***Taula entitat\_bancaria***

Aquesta taula és la representació física de la classe EntitatBancaria on la clau primària es codiBanc, en aquest cas també es podria haver optat per fer una clau doble entre numEntitat i numOficina, però per facilitat s'ha optat per una clau nova, com codiBanc.


















Column Name	Datatype	NOT NULL
 codiBanc	 INT(3)	
 numEntitat	 INT(4)	
 numOficina	 INT(4)	
 nomEntitat	 VARCHAR(100)	
 nomOficina	 VARCHAR(100)	
 domicili	 VARCHAR(45)	
 poblacio	 VARCHAR(25)	
 pais	 VARCHAR(20)	
 personaContacte	 VARCHAR(20)	
 telefon1	 INTEGER	
 telefon2	 INTEGER	
 fax	 INTEGER	

### ***Taula entitatcomercial***

Taula que representa la classe EntitatComercial del model de classes. Degut a que és una subtaula d'entitat les claus primàries seran codiEntitat i codiEntitatComercial, mentre que codiEntitatComercial també en serà clau forana de la taula.

Column Name	Datatype	NOT NULL
 codiEntitat	 INT(4)	
 codiEntitatCome...	 INT(10)	
 nomEntitatCome...	 VARCHAR(45)	
 www	 VARCHAR(45)	
 PersonaContacte	 VARCHAR(25)	
 fax	 INTEGER	
 descompteCom...	 DOUBLE	
 descomptePaga...	 DOUBLE	
 iva	 DOUBLE	
 re	 DOUBLE	
 riscNoPagament	 DOUBLE	
 recarregFinancer	 DOUBLE	
 observacions	 VARCHAR(250)	











***Taula facturacompra***

Column Name	Datatype	NOT NULL
 codi	 INTEGER	
 numeroFactura...	 VARCHAR(45)	
 dataEntrega	 DATETIME	
 dataPagament	 DATETIME	
 import	 DOUBLE	
 codiProveidor	 INTEGER	

Aquesta taula representa amb el mateix nom a la classe FacturaCompra del diagrama de classe, on la clau primària es el codi i la clau forana fruit de la relació (1..N) entre la facturacomanda i el proveïdor.























***Taula familia***

Aquesta taula representa la classe Familia del model de classes on la clau primària és el codiFamilia i la clau foranà es idFamilia que fa referència a la mateixa classe per la seva relació (1..N).

Column Name	Datatype	NOT NULL
 codiFamilia	 INT(3)	
 nomFamilia	 VARCHAR(20)	
 descripcio	 VARCHAR(200)	
 idFamilia	 INTEGER	
 nomFamiliaSuper	 VARCHAR(250)	









***Taula liniadocument***

Taula que representa totes les línies dels diferents documents, on la clau primària és el codi. Les dues claus foranes que trobem són codiArticle i document on codiArticle fa referència a la relació (1..N) entre liniadocument i article i document fa referència a la relació (1..N) entre document i liniadocument.

Column Name	Datatype	NOT NULL
 codi	 INTEGER	
 codiArticle	 INTEGER	
 preu	 DOUBLE	
 descompte	 DOUBLE	
 quantitat	 DOUBLE	
 iva	 DOUBLE	
 ImportTotal	 DOUBLE	
 document	 INTEGER	

***Taula proveïdor***

Aquesta taula que representa la classe treballador de model de classes i com en ella també es una subclasse de entitatcomercial. Les claus primàries d'aquesta taula són codiEntitatComercial i codiProveïdor on codiEntitatComercial també en serà clau forana de la taula.

Column Name	Datatype	NOT NULL
 codiEntitatCome...	 INT(4)	
 codiProveïdor	 INT(4)	
 tipusAtencio	 VARCHAR(10)	



**Taula treballador**

Aquesta taula que fa referència a la classe treballador del model de classes, té com a clau primària codiEntitat fruit de la relació d'herència que tenen Entitat i treballador, es per això que codiEntitat de la mateixa manera que codiTreballador són claus primàries de la taula, i el mateix temps codiEntitat serà clau forana de la taula.

Column Name	Datatype	NOT NULL
codiEntitat	INT(4)	
codiTreballador	INT(3)	
hores	INT(2)	
sou	INT(4)	
llocOcupat	VARCHAR(45)	

**Taula treballador\_zona**

Aquesta taula és fruit de la relació (N..M) entre zona i treballador, per la qual cosa codiTreballador, per consegüent codiEntitat també serà clau primària, i codiZona. I cada una d'elles també serà clau forana de la taula.

Column Name	Datatype	NOT NULL
codiTreballador...	INT(3)	
codiEntitat	INT(3)	
codiTreballador	INT(3)	
codiZona	INT(3)	

**Taula zona**

Aquesta taula representa la classe Zona del diagrama de classes on la clau primària és codiZona. Aquesta taula no contindrà claus foranes.

Column Name	Datatype	NOT NULL
codiZona	INT(3)	
nomZona	VARCHAR(45)	
descripcioZona	VARCHAR(45)	

## **4.6 Especificacions de construcció**

### **4.6.1 Especificacions de l'entorn de construcció.**

Per a la construcció del sistema d'informació cal definir quin és l'entorn.

#### **Software**

Per la construcció del sistema es farà sobre el sistema operatiu Microsoft Windows XP Professional, ja que les eines que s'utilitzaran per el desenvolupament de construcció de l'aplicació s'executen sobre aquest sistema.

#### **Hardware**

: L'equip emprat usualment pel desenvolupament de sistema és:

- Processador Intel(R) Core(TM) 2 CPU
- 1 Gb de memòria RAM
- 140 Gb de disc dur.
- 128 Mb de tarja gràfica.
- Grabadora de CD-ROM

#### **Llenguatge de programació**

El llenguatge de programació emprat ha estat el Java, que ha estat el llenguatge més utilitzat durant la carrera universitària.

#### **Eines de desenvolupament**

Un cop escollit els llenguatges de programació, caldrà triar quines eines s'utilitzen per el seu desenvolupament. Les eines ha utilitzar són Eclipse i NetBeans. NetBeans és la eina de suport que ajudarà a crear les pantalles, ja que tot hi que Eclipse també té aquesta eina, NetBeans té un entorn més amigable per realitzar aquest tipus de feina.

***Sistema gestor de bases de dades***

Per l'elecció del sistema de base de dades s'ha escollit MySQL primer per que és lliure d'impostos i dins de les altres possibilitats que hi havia com Microsoft SQL Server, Oracle i Microsoft Access XP, MySQL ha estat el sistema amb el que encara no he treballat mai.

## 5 Construcció del sistema d'informació

Les activitats que es que és veuran en aquest capítol són les següents:

- Preparació de l'entorn de generació i construcció.
- Generació del codi dels components.
- Execució de les proves.
- Elaboració del manual d'usuari.
- Definició de la formació de l'usuari final

### 5.1 Preparació de l'entorn de generació i construcció.

Primer de tot s'haurà de preparar el sistema gestor de base de dades emprat és MySQL Server 6.0, amb aquesta aplicació es generaran totes les taules, claus primàries i relacions necessàries, que s'han analitzat en el capítol de disseny físic de les dades.

Tot seguit s'instal·larà l'entorn de desenvolupament Eclipse i llenguatge de programació Java. En aquest cas s'hauran d'importar les llibreries necessàries per la realització del projecte. Per la creació ràpida de la interfície gràfica amb Java es necessitarà instal·lar el programa NetBeans.

### 5.2 Generació de codi de components

Per la generació de codi es farà unitàriament per cada cas d'ús definit en el disseny i es faran les respectives proves.

En la generació del codi a part de fer-ho cas per cas s'ha pensat mantenir l'ordre de crear primer el codi per la capa d'entitats que intervenen en el cas, tot seguit la capa de la base de base de dades, després per la capa de control i funcions conjuntament amb la part de l'interfície gràfica.

### **5.3 Elaboració de proves**

Tal hi com s'ha dit anteriorment cada joc de proves s'executen unitàriament en la realització de cada cas d'ús.

Un cop realitzades les proves per a cada cas d'ús i veien que no s'han creat errors, es passa a fer el joc de proves integrant tots els casos d'ús per tal que no hi hagin interferències entre ells.

En les proves fetes en l'aplicació se'n realitzen comprovacions per tal que els resultats obtinguts siguin els correctes i no hi haguí errors que puguin repercutir a l'usuari final.

### **5.4 Elaboració dels manuals d'usuari**

En acabar tot el procés de generació de codi, un cop acabada l'aplicació s'ha de passar a crear un manual d'usuari, per tal d'explicar quin és el funcionament del programa. D'aquesta manera es donarà suport al futur usuari de l'aplicació amb la resolució de qualsevol dubte que tingui en el funcionament de l'aplicatiu.

El manual ha de contenir l'explicació de com l'usuari a de preparar la seva màquina per tal que l'aplicació funcioni correctament.

L'explicació del funcionament de l'aplicació a partir dels seus menús, fins arribar a les pantalles, comentant els diàlegs que poden aparèixer.

### **5.5 Definició de la formació de l'usuari final**

Per tal de poder utilitzar l'aplicació l'usuari final tan sols haurà de tenir els coneixements mínims informàtics, com son els de encendre l'ordinador, realitzar una instal·lació i navegació. I finalment amb l'ajuda del manual que se li proporciona amb l'aplicació n'ha de tenir suficient.

## 6 Incorporació del sistema a l'entorn final

En aquest capítol, es detallaran les activitats següents:

- Incorporació del sistema a l'entorn final.
- Realització de proves d'implantació
- Realització de proves d'acceptació del sistema.

### 6.1 Incorporació del sistema a l'entorn final.

Abans de realitzar la instal·lació de l'aplicació s'haurà de comprovar que l

### 6.2 Realització de proves d'implantació

Un cop feta d'incorporació del sistema a l'entorn final caldrà fer tot una seria de proves per tal de comprovar que la funcionalitat de l'aplicació dins de l'entorn i que sigui correcte d'acord amb el que s'havia especificat inicialment

### 6.3 Realització de proves d'acceptació del sistema.

Com que no hi ha un usuari final establert, la realització de proves que s'ha fet per l'acceptació del sistema són les mateixes que s'han anat realitzant durant el procés de realització del projecte.

En finalitzar el projecte s'han pogut arribar a un seguit de millores que es podria fer, per tal de fer encara més àgil la utilització i la gestió de les diferents funcions.

- Els proveïdors es podrien classificar segons el tipus d'activitat que realitzen cap a l'empres, es a dir, si són proveïdors d'articles, de serveis, de combustibles,...
- Els articles es podrien diferenciar entre productes o serveis.
- A l'hora d'entrada i sortida de productes fer-ne un seguiment es a dir guardar un historial.
- Crear una interfície una mica més amigable.
- Augmentar el número de consultes a poder realitzar, es a dir una més flexibilitat per realitzar consultes i filtres.

## Conclusions

### **Objectius assolits.**

Un cop acabat el projecte, per treure'n unes conclusions de com ha anat tota la seva elaboració i el resultat obtingut el millor que es pot fer és mirar si els objectius que s'havien marcat al principi han estat aconseguits satisfactòriament.

En aquest sentit, podem dir que hem assolit els objectius previstos. S'ha desenvolupat una aplicació destinada a empesaris autònoms que automatitza les tasques més habituals per a aquest perfil d'usuaris: gestió de compres i vendes, facturació, etc. Considerem, a més, que hem assolit l'objectiu de fer una aplicació fàcil d'utilitzar per a usuaris no informàtics.

### **Què he après amb aquest projecte?**

D'entrada cal dir que aquest projecte ha suposat poder realitzar una aplicació de gran volum en totes les etapes del seu desenvolupament, i seguint una metodologia.

En la creació d'aquest projecte he posat en practica gairebé la majoria dels coneixements adquirits al llarg de la carrera com ara l'anàlisi i disseny de bases de dades, l'enginyeria del software, la programació en Java, etc.

A part, en el transcurs de la realització d'aquest projecte he aprofitat per introduir-me en la creació d'interfície gràfica amb Java, ja que, durant la carrera, l'únic que havia arribat a fer era una interfície molt senzilla amb Visual Basic, i crec que un programa fet amb una bona interfície gràfica és més atractiu. Una altra cosa que cal remarcar és l'aprofundiment que he fet en les llibreries del llenguatge de programació Java ja que, a part d'aprendre sobre la llibreria de creació de pantalles, també he descobert i utilitzat d'altres que anteriorment no havia fet servir.

Una altra eina que he pogut conèixer gràcies a aquest projecte ha estat el sistema gestor de base de dades MySQL que tampoc havia utilitzat abans i que m'ha sorprès per les eines que ofereix per a la gestió de la base de dades.

També vull destacar la posada en pràctica de l'arquitectura de programació per capes, la qual m'ha ajudat molt a l'hora de poder desenvolupar el projecte, ja que ajuda a entendre millor què està passant a cada moment i fa que el codi quedi més ben ordenat.

A part dels aspectes més tècnics ja citats, voldria destacar altres aspectes més transversals que he anat assolint al llarg del desenvolupament del projecte, com ara:

- consulta de bibliografia científico-tècnica i de manuals
- documentació completa del projecte
- presentacions orals

Finalment, vull destacar i agrair l'ajut que he tingut per part del meu director de projecte, la col·laboració del meu pare en els dubtes empresarials que han anat apareixent al llarg del desenvolupament, i també l'ajut obtingut en els fòrums de la xarxa.



## **Bibliografia**

### **Edició i desenvolupament.**

- **Eclipse**
- **NetBeans**

### **Disseny i desenvolupament de l'anàlisi en diagrames**

- **Microsoft Office Visio 3.0**

### **Processador de textos**

- **Microsoft Office 2003**

### **Manuais de programació i eines.**

- **API de Java** : <http://java.sun.com/j2se/1.5.0/docs/api/>
- **Pagina de MySQL**: <http://www.mysql.org>